AD A 279640

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 1994	3. REPORT TYPE AND DATES COVERED Final 7 May 1993 – 17 December 1993

4. TITLE AND SUBTITLE

Modeling Platform Dynamics and Physiological Response to Short Arm Centrifugation

5. FUNDING NUMBERS

C - F41624-93-C-6011
PE - 65502F
PR - 3005
TA - CT
WU - 3B

6. AUTHOR(S)

David J. Pancratz
John B. Bomar, Jr.
James H. Raddin, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Biodynamic Research Corporation
9901 IH 10 West, Suite 1000
San Antonio, TX 78230

8. PERFORMING ORGANIZATION REPORT NUMBER

F41624-93-C-6011

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Armstrong Laboratory (AFMC)
Crew Systems Directorate
Crew Technology Division
2504 D Drive, Suite 1
Brooks Air Force Base, TX 78235-5104

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AL/CF-TR-1994-0025

11. SUPPLEMENTARY NOTES

Armstrong Laboratory Technical Monitor: Larry J. Meeker, (210) 536-2742. This research was conducted under th3 Small Business Innovation Research (SBIR) program as a Phase I effort.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unumited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

An analysis is presented for the operation of a short radius centrifuge on an orbiting platform. A review of literature confirms that such a centrifuge could be useful in ameliorating the effects of microgravity deconditioning of astronauts. Our analysis indicates that operation of the centrifuge could generate potentially destabilizing forces and moments. Several ideas for reducing or eliminating the forces and moments are discussed. The use of pedals to drive the centrifuge and exercise the rider is also described. Excess power from pedalling could be stored as electrical energy. A mathematical model of the human cardiovascular system indicates that centrifugation at a short radius would indeed stimulate the heart, even though there is a significant acceleration gradient from head to toe. Consideration of the disorienting effects of short radius centrifugation reveals that discomfort originating in the otoliths from the lack of gravity would probably be alleviated; however, the angular motion of the centrifuge would likely cause disorientation or discomfort through the semicircular canals.

14. SUBJECT TERMS

Acceleration physiology
Cardiovascular modeling
Microgravity countermeasures
Short arm centrifuge
Torque balancing mechanism

15. NUMBER OF PAGES
182

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

CONTENTS

FIGURES

TABLES

ACKNOWLEDGMENTS

The members of the research team express their deep appreciation to the many individuals making contributions to this research program. Without the timely assistance given so generously by everyone, this research program and this report could not possibly have been accomplished.

A special thanks is given to Mr. Larry Meeker as the USAF Project Engineer in the Crew Systems Directorate, Armstrong Laboratory, Human Systems Division, for his support, management and leadership during this research program.

Finally, the research team gives a very special thank you to the management and support staff at BRC. Without the superb efforts of Ms. Karen Lindley, the editing and publication of this research program could not have been accomplished effectively. Appreciation is also given to Mr. John Martini and Mr. Bob Carlin for the extensive illustration support, to Ms. Pat Perret for the technical library support, and to Mr. Tom Kingery for contract management.

DESCRIPTION OF SYMBOLS AND NOTATION

Vectors: Bold face type indicates a vector quantity. The subscript represents a reference point. The superscript represents a reference coordinate frame. For example,

$$\mathbf{a}^{C}_{A/B} ,$$

should be interpreted as the acceleration vector of the point A relative to point B expressed in the C coordinate frame. The only exception to this rule is in the case of unit vectors or x, y, and z projections of a vector. In these cases, the coordinate frame will appear as a subscript.

Coordinate Frames (used as subscripts or superscripts):

L	-	Local Vertical Local Horizontal (LVLH)
SS	-	Space Shuttle
C	-	Centrifuge
P	-	Crewmember (center of mass)
H	-	Pilot Head
O	-	Head Otolith
T	-	Tangential in centrifuge frame
R	-	Radial in centrifuge frame
x	-	Denotes X-axis projection of vector in any frame
y	-	Denotes Y-axis projection of vector in any frame
z	-	Denotes Z-axis projection or vector in any frame

Unit Vectors:

i	-	X-axis
j	-	Y-axis
k	-	Z-axis

Symbols:

$\mathbf{a}$	-	acceleration vector [length·sec^{-2}]
AFR	-	afferent firing rates [impulse·sec^{-1}]
C	-	cosine of an angle
E	-	energy [force·length]
$\mathbf{g}_0$	-	Earth's gravitational acceleration [9.81 m·sec^{-2}]
G_0	-	Earth's gravitational acceleration in units of g_0
G	-	acceleration vector normalized expressed in multiples of g_0
$\mathbf{H}$	-	angular momentum [mass·length^{2}·sec^{-1}]
I	-	mass moment of inertia [mass·length^{2}]
K	-	control compensator gain
L	-	3 x 3 coordinate system rotation matrix
M	-	moment [force·length]
P	-	Power [energy·sec^{-1}]
r	-	centrifuge radius [length]
$\mathbf{r}$	-	position vector [length]
s	-	Laplace transformation variable
S	-	sine of an angle
$\mathbf{v}$	-	velocity vector [length·sec^{-1}]
t	-	time [sec]
T_L	-	canal long time constant [sec]
α	-	angular acceleration vector [deg·sec^{-2}]
γ	-	angular position vector [deg]
δ	-	incremental change or perturbation of a quantity
ϵ	-	error value [any unit]
τ	-	dynamic system time constant [sec]
ω	-	angular velocity vector [deg·s^{-1}]

Time Derivatives:

˙	-	First time derivative
¨	-	Second time derivative

SECTION 1.0

1.0 Summary.

Biodynamic Research Corporation (BRC) of San Antonio, TX, completed an SBIR Phase I project to study the concept of a short radius centrifuge for use in a microgravity environment. The impetus for this project was the desire to further studies of a conceptual short radius centrifuge as a mechanism to ameliorate the effects of microgravity deconditioning.[1] From an orbital dynamics consideration, motion of the centrifuge and riders can cause destabilizing forces and moments to be applied to the orbiting platform. Physiologically, centrifugation may exacerbate the discomfort associated with motion in microgravity and create an increased potential for disorientation. Furthermore, although research has shown short radius centrifugation to be tolerable on earth, there have been no studies of the effect of centrifugation on the cardiovascular system in microgravity.

To quantify the dynamic effects of centrifugation on an orbital platform, we considered as a baseline configuration a 200 cm radius centrifuge to be operated in a pressurized enclosure of the space shuttle payload bay. The device was considered to be aligned with a spin axis parallel to the shuttle x-axis, which points from the nose towards the engines. To minimize the angular velocity of the centrifuge, the riders are assumed to be positioned with the top of their heads approximately 100 cm from the center of the device.

Parametric studies of the centrifuge with a single rider indicate the dynamic imbalance from the rotating mass will cause the shuttle center of mass to move in a circular path with diameter of 0.1 cm and cause an angular displacement on the order of 0.01° from the shuttle x-axis. Although the force and moment caused by the imbalance are large in magnitude, they result in accelerations to the shuttle, approximately 10^{-3} G, in a frequency range that has been tolerated on other spacecraft. Methods for reducing and eliminating the imbalance are described. The simplest solution is to place a second rider opposite the first on the centrifuge. This reduces the dynamic imbalance effect by 50% or more, however an additional mechanism is required to reduce the imbalance further. We propose an automated mechanism with a moveable countermass to continually balance the centrifuge. The addition of this device should eliminate most of the shuttle oscillation caused by the imbalance.

Centrifugation with a single rider will also produce a drive moment, from angular acceleration of the crewmember inertia, and gyroscopic moment, from the motion of the centrifuge spin axis due to shuttle maneuvering and orbiting. Compensation for these moments may be of more concern than the imbalance moment because they potentially act over long durations. To eliminate these moments, we propose a counter-rotating inertia similar to conventional spacecraft reaction wheels. To minimize the mass, the counter-rotating inertia could be rotated at a faster rate than the centrifuge.

We also considered the use of pedalling to drive the centrifuge. Sustainable power and torque outputs from a single rider should be adequate to drive the centrifuge, either through a direct mechanical connection or by conversion to electrical power.

Dynamic mathematical models of the vestibular system were used to determine how the rider would be affected by centrifugation. Literature from other excursions in space indicate microgravity causes normal small translations of the head to cause discomfort through stimulation of the otoliths. The radial acceleration of the centrifuge may help alleviate this discomfort. The angular motion of the centrifuge, on the other hand, will create the potential for disorientation or discomfort by stimulation of the semicircular canals. Of particular concern is the potential for the cross-coupling illusion, which may be more prevalent for riders on the centrifuge than for high performance aircraft pilots.

To be of use in ameliorating the effects of microgravity on the cardiovascular system, the short radius centrifuge must be able to create enough fluid static pressure difference in the rider to stimulate the cardiovascular system. A mathematical model of the human cardiovascular system indicates that centrifugation at a short radius in space would indeed stimulate the heart, even though there is a significant acceleration gradient from head to toe. The predicted static pressures throughout the cardiovascular system for centrifugation in microgravity are similar to the pressures of centrifuge riders on earth.

SECTION 2.0

2.0 Introduction

This final report concludes an effort by Biodynamic Research Corporation (BRC) of San Antonio, TX to conduct a parametric study of the use of a short arm centrifuge for space applications. The effort was conducted under Contract #F41624-93-C-6011 through the Armstrong Laboratory and is entitled "Modelling Platform Dynamics and Physiologic Response to Short Arm Centrifugation."

The study emanates from the desire to use centrifuges as a countermeasure to prevent the physiologic deconditioning of astronauts exposed to long term microgravity environments. An Air Force parametric design study for a small radius centrifuge revealed such a centrifuge could fit on the NASA Space Shuttle and provide artificial gravity and exercise for crewmembers.[1] An SBIR Phase I project was awarded to BRC to investigate several physiological effects of short radius centrifugation, as well as assess the impact of the centrifuge motion on the space shuttle dynamics. Specifically, the objectives of the Phase I project are listed below:

(1) To develop a kinematic model of the acceleration produced by a short radius centrifuge.

(2) To develop a simple mathematical model of the steady state cardiovascular response to short radius centrifugation.

(3) To develop a mathematical model of the crewmember vestibular stimulus and response produced by short radius centrifugation.

(4) To assess the effects of centrifugation on the orbiting platform and consider methods for negating destabilizing forces and moments.

Biodynamic Research Corporation performed all of the analysis for this contract. This final report documents and summarizes the work generally according to the objectives outlined above. Chapter 3 of this final report provides background on physiologic deconditioning in microgravity and the use of centrifuges for space application. Chapter 4 describes the kinematics and dynamics of a space-based centrifuge, typical destabilizing forces and moments transmitted to the orbiting platform, and methods for eliminating destabilizing forces and moments. Finally, Chapter 5 describes some of the physiologic effects of centrifugation in space, including simulations of the theoretical vestibular response and cardiovascular effects of centrifugation.

SECTION 3.0

3.0 Technical Background and Literature Review

Much research has been conducted in the areas of physiologic deconditioning due to exposure to microgravity and the possible ameliorating effects of centrifugation. In this chapter, we review some of the physiologic effects and briefly trace the history and technical literature discussing space-based centrifuges.

3.1 Microgravity Deconditioning.

Humans and animals have evolved adaptation to earth's gravity to maintain physiologic homeostasis.[2] Prolonged exposure to microgravity, such as that experienced by astronauts, has several negative physiologic effects collectively termed deconditioning. Some of the most significant effects include:[3]

- fluid shift and decreased plasma volume;
- orthostatic intolerance;
- reduced tolerance to increased $+G_z$ acceleration;
- negative calcium balance, resulting in the loss of bone;
- possible immunosuppression; and
- muscle atrophy.

The musculoskeletal system is susceptible to changes in acceleration loading. Studies in microgravity have shown a reduction in skeletal muscle mass in long duration exposures, which can be partially offset by physical exercise.[2] Bone mass losses are primarily losses in mineral content, which changes the mechanical properties of bones. The magnitude of muscle atrophy and bone demineralization has been shown to be directly related to the length of microgravity exposure.[4]

Orthostatic intolerance is regarded as a potentially serious cardiovascular consequence of microgravity exposure. The intolerance is attributable to the absence of hydrostatic pressure gradients in microgravity.[4] The lack of a head-to-foot gradient causes a significant shift in fluid from the lower body to the head and chest after several days in space. Other physiologic responses reduce the total blood volume to accommodate the fluid shift, and a new circulatory homeostasis is achieved. When the cardiovascular system is once again exposed to gravity, it is unable to compensate properly for the pressure gradient.

Although these effects may not necessarily jeopardize a mission in space, orbital experiences and postflight medical examinations of astronauts have suggested deconditioning may interfere with performance during reentry and subsequent life on earth.[4,5] It is important that humans be able to survive a long duration exposure to microgravity and return safely to earth's gravity.

3.2 Centrifuge for Space Applications.

Since the cause of physiologic deconditioning in microgravity is the absence of gravity, the most obvious means for reducing the effects is to simulate the gravitational loading found on earth. Other interventions have been tried; for example, in-flight exercise regimens have been shown to provide protection against different aspects of deconditioning.[4] Lower Body Negative Pressure (LBNP) has also been used to improve post-flight orthostatic intolerance.

Even more useful for preventing deconditioning might be the radial acceleration created by a centrifuge.[6] Scientific evidence indicates that periodic stimulation of the body by acceleration causes a physiologic response that persists after the stimulation.[2] Numerous studies advocate centrifugation as a countermeasure to space deconditioning.[1,3,7,8,9,10] Acceleration magnitude and duration appear to be interactive; increasing the acceleration magnitude may reduce the time required to ameliorate deconditioning.[2]

In a spacecraft, where size is a prime consideration, a centrifuge has to have a small radius. Since the acceleration at a point on the centrifuge is dependent on the distance from the center (discussed more fully in Chapter 4), an acceleration gradient exists across the crewmember. The cardiovascular effects of this gradient are discussed in Chapter 5.

SECTION 4.0

4.0 Platform and Centrifuge Dynamics

The motion of the short radius centrifuge and riders has the potential to transmit destabilizing forces and moments to the orbiting platform. For this report, we have assumed the orbiting platform is the space shuttle. In Sections 4.1 and 4.2 convenient coordinate frames are identified and the three-dimensional dynamics of the centrifuge and shuttle are described. Section 4.3 quantifies the reaction forces and moments that result from centrifugation, and the results of a shuttle dynamic simulation are presented. In Section 4.4, the use of rider pedalling for aerobic exercise and as a means to drive the centrifuge is discussed. Methods for eliminating mass imbalance and compensating for centrifugation and pedalling moments are described in Section 4.5. Finally, Section 4.6 summarizes the results of Chapter 4 and presents some concept designs for a pedal-powered space centrifuge.

4.1 Coordinate Frames and Kinematics.

Analysis of the motion of the centrifuge, space shuttle, and crewmember requires the identification and definition of appropriate coordinate frames. Four coordinate frames are used for the shuttle, centrifuge, and crewmember kinematics: (1) an orbital Local Vertical Local Horizontal (LVLH) frame; (2) a space shuttle fixed frame; (3) a centrifuge fixed frame; and, (4) a crewmember fixed frame. Each frame is described in detail below and depicted in Figure 4-1.

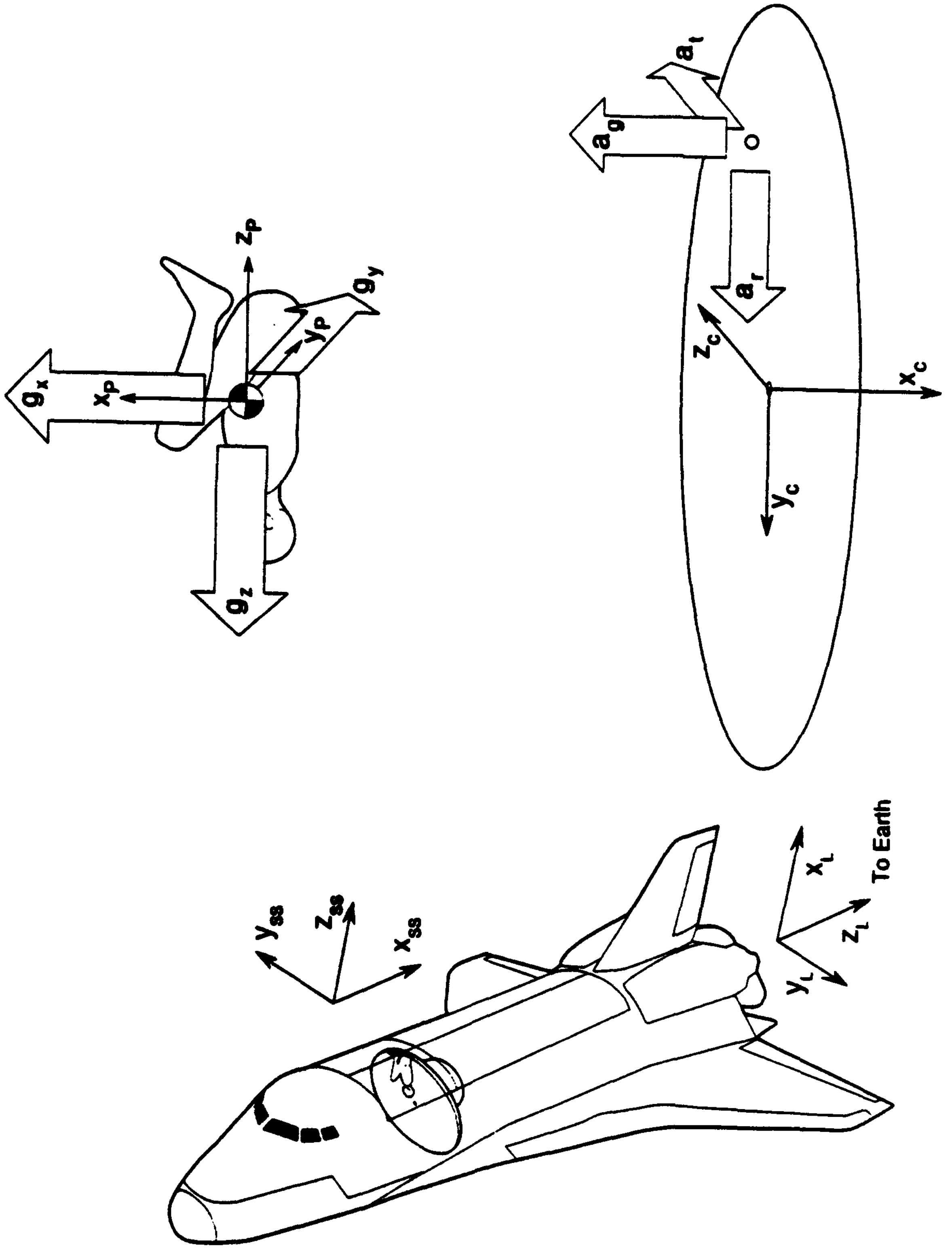
(1) LVLH. The Local Vertical Local Horizontal coordinate frame is defined by the orbital plane of the space shuttle and moves with the shuttle. The LVLH z-axis always points towards the center of the earth. The x-axis always points in the direction of tangential velocity of the shuttle. The y-axis is orthogonal to the x and z axes and represents a unit normal to the orbital plane of the shuttle. This frame is denoted by the letter L.

(2) Space Shuttle Fixed. This coordinate system is fixed to the space shuttle center of mass and moves with the shuttle. The x-axis points from the nose to the engines of the shuttle. The z-axis is directed normal to the shuttle wings with the top of the shuttle being the positive direction. The y-axis is directed from the shuttle out towards the right. This frame is denoted by the letters SS.

(3) Centrifuge. This coordinate frame is fixed to the center of the centrifuge. The x-axis is the spin axis, and the y and z axes are selected to be parallel to the shuttle axes. This frame is denoted by the letter C.

(4) Crewmember. This coordinate frame is fixed to the center of mass of a crewmember on the centrifuge. The z-axis points from head to toe, the x-axis points forward, and the y-axis points out the right side of the subject. This frame is denoted by the letter P.

Figure 4-1. Centrifuge and Shuttle Coordinate Systems



4.2 Platform and Centrifuge Kinematics and Dynamics.

To simplify the analysis of the shuttle and centrifuge motion, BRC made several assumptions:

(1) The orbital velocity, eccentricity, altitude, and other aspects of the shuttle orbit were not considered in the analysis.

(2) Deviations in the shuttle orientation were considered small enough that the original unrotated axes were used to express the motion.

(3) Movement of the centrifuge and riders causes a negligible change in the center of mass and inertia of the shuttle. This alleviates the need to recompute the shuttle mass properties for centrifuge simulations and parametric studies.

(4) Space shuttle and centrifuge mass properties information were obtained from an Air Force publication.[1] Human data was obtained from an anthropometry source.

(5) Unless otherwise noted, dynamic and kinematic vector quantities are assumed to be with respect to the space shuttle coordinate frame.

(6) The existing momentum management system of the space shuttle was not considered; we assumed the shuttle has no source for angular momentum storage or resistance other than the motion of the centrifuge and the shuttle inertia.

The total angular velocity and acceleration of the short arm centrifuge due to angular motion of the shuttle and centrifugation can be found from:

$$\boldsymbol{\omega}_C = \boldsymbol{\omega}_{SS} + \boldsymbol{\omega}_{C/SS} \ , \qquad [4\text{-}1]$$

and

$$\boldsymbol{\alpha}_C = \boldsymbol{\alpha}_{SS} + \boldsymbol{\alpha}_{C/SS} + \boldsymbol{\omega}_{SS} \times \boldsymbol{\omega}_C \ . \qquad [4\text{-}2]$$

Then, the velocity and acceleration of a crewmember, P, on the short arm centrifuge can be computed with the following expressions:

$$\boldsymbol{v}_P = \boldsymbol{v}_{SS} + \boldsymbol{\omega}_C \times \boldsymbol{r}_{P/SS} \ , \qquad [4\text{-}3]$$

and

$$\boldsymbol{a}_P = \boldsymbol{a}_{SS} + \boldsymbol{\alpha}_C \times \boldsymbol{r}_{P/SS} + \boldsymbol{\omega}_C \times (\boldsymbol{\omega}_C \times \boldsymbol{r}_{P/SS}) \qquad [4\text{-}4]$$

Assuming that the motion of the shuttle is negligible compared to that of the centrifuge, the magnitude of the radial and tangential acceleration of a subject whose center of mass is located at the point P can be approximated by:

$$a_R = \omega^2_{C/SS} r_{P/C} \ , \qquad [4\text{-}5]$$

and

$$a_T = \alpha_{C/SS} r_{P/C} \ . \qquad [4\text{-}6]$$

Because the effect of gravity cn an orbiting centrifuge is negligible, crewmembers can be arbitrarily placed on the centrifuge while it is motionless without static balance requirements. Once the centrifuge is accelerated, however, dynamic imbalance of the rider masses can impart a force and moment to the shuttle. The reaction force generated by centrifuge dynamic imbalance is quantified.by:

$$F_{imb} = -\sum (m_{P_i} a_{P_i}) \ . \qquad [4\text{-}7]$$

Since this is assumed to be the only disturbing force on the shuttle, translational motion of the shuttle center of mass can then be computed from:

$$a_{SS} = \frac{F_{imb}}{m_{ss}} \ . \qquad [4\text{-}8]$$

If the centrifuge is not located at the space shuttle center of mass, the force due to imbalance applied at a distance will create an imbalance moment on the shuttle:

$$M_{imb} = r_{C/SS} \times F_{imb} \ . \qquad [4\text{-}9]$$

Angular acceleration of the centrifuge and precession of the centrifuge while spinning due to shuttle motions will also transmit disturbing moments to the shuttle. Consider that the motion of the centrifuge is governed by:

$$\sum M_C = \left(\frac{d|H_C|}{dt}\right) + \omega_{SS} \times H_C \ , \qquad [4\text{-}10]$$

where

$$H_C = I_C \omega_{C/SS} \ , \qquad [4\text{-}11]$$

and the inertia of the centrifuge about the x_C axis can be computed from:

$$I_C = I_{Cfuge} + \sum \left(I_{p_i} + m_{p_i} r_{p_i}^2 \right) \ , \qquad [4\text{-}12]$$

where p_i represents each of the crewmembers.

Substituting Equation 4-11 into 4-10 yields an expression for the moments created by the centrifuge:

$$\sum M_C = I_C \alpha_{C/SS} + \omega_{SS} \times I_C \omega_{C/SS} \ . \qquad [4\text{-}13]$$

An alternate form for Equation 4-13 is:

$$\sum M_C = M_{drv} + M_{gyro} \ , \qquad [4\text{-}14]$$

where

$$M_{drv} = I_C \alpha_{C/SS} \ , \qquad [4\text{-}15]$$

and

$$M_{gyro} = \omega_{SS} \times I_C \omega_{C/SS} \ . \qquad [4\text{-}16]$$

Angular motion of the space shuttle due to the mass imbalance moment and centrifuge drive and gyroscopic moments is computed from:

$$\sum M_{SS} = \dot{H}_{SS} \ , \qquad [4\text{-}17]$$

where

$$\dot{H}_{SS} = \frac{d}{dt}(I_{SS} \omega_{SS}) = I_{SS} \alpha_{SS} \ . \qquad [4\text{-}18]$$

Substituting Equations 4-15, 4-16, and 4-18 into 4-17 yields the final expression for angular motion of the space shuttle:

$$\alpha_{SS} = I_{SS}^{-1} (M_{imb} - M_{drv} - M_{gyro}) \ . \qquad [4\text{-}19]$$

Note in Equation 4-19 that the drive and gyroscopic moments appear as negative quantities because it is the equal and opposite *reaction* moments that act on the shuttle.

4.3 Parametric Studies and Simulations.

Using the analysis of Section 4.2, the destabilizing forces and torques from centrifuge motion are quantified in this section. First, it is necessary to summarize the anthropometry of typical astronauts and the mass properties and dimensions of the shuttle and centrifuge. With these values, expressions from Section 4.2 are used to describe motion of the centrifuge and shuttle.

4.3.1 Crew, Centrifuge, and Shuttle Dimensions and Properties.

To assist in selecting the size of the centrifuge, the relevant anthropometric data for astronauts are summarized in Table 4-1.[11]

The maximum diameter of the space shuttle crew compartmen*. is approximately 130 cm, which would permit a centrifuge with arm length of only 65 cm.[1] From Table 4-1, it is evident that even the shortest astronaut could not lie in a supine position on a centrifuge with 65 cm arm. The shuttle payload bay, however, is much larger than the crew compartment -- the nominal diameter is 457 cm.[1] A centrifuge with maximum arm length of approximately 225 cm could be used in the payload bay. For the 95 percentile male, the subject's head could be positioned a maximum of approximately 125 cm from the center of the centrifuge. For most of the parametric studies and simulations in this report, the crewmember's head is positioned 100 cm from the center of the centrifuge. As in the Air Force study, we will assume the centrifuge is located in a pressurized enclosure in the shuttle payload bay.[1]

For the dynamic motion analysis, mass properties for the shuttle were obtained from the Meeker and Isdahl study.[1] The total mass of the shuttle is estimated to be 98,122 kg, and the inertia matrix about the center of mass is:

$$I_{SS} = \begin{bmatrix} 1{,}453{,}255 & -7{,}544 & -335{,}642 \\ -7{,}544 & 10{,}077{,}164 & -10{,}259 \\ -335{,}642 & -10{,}259 & 10{,}346{,}577 \end{bmatrix} kg{\cdot}m^2 \; . \qquad [4\text{-}20]$$

A typical single centrifuge arm could be fabricated from two pieces of 200 cm long aluminum tube, with fabric stretched between to support the rider and straps to hold him in place. If the mass of the structure is approximated at 12.5 kg and center of mass assumed to be 100 cm from the centrifuge center, the moment of inertia of each arm will be 12.5 kg·m^2. A two-arm centrifuge, which is discussed frequently in subsequent sections of this report, would have a combined center of mass at the centrifuge axis, total mass of 25 kg, and moment of inertia of 25 kg·m^2. A schematic of a conceptual centrifuge configuration with single rider is displayed in Figure 4-2.

Figure 4-2. Centrifuge with Single Rider

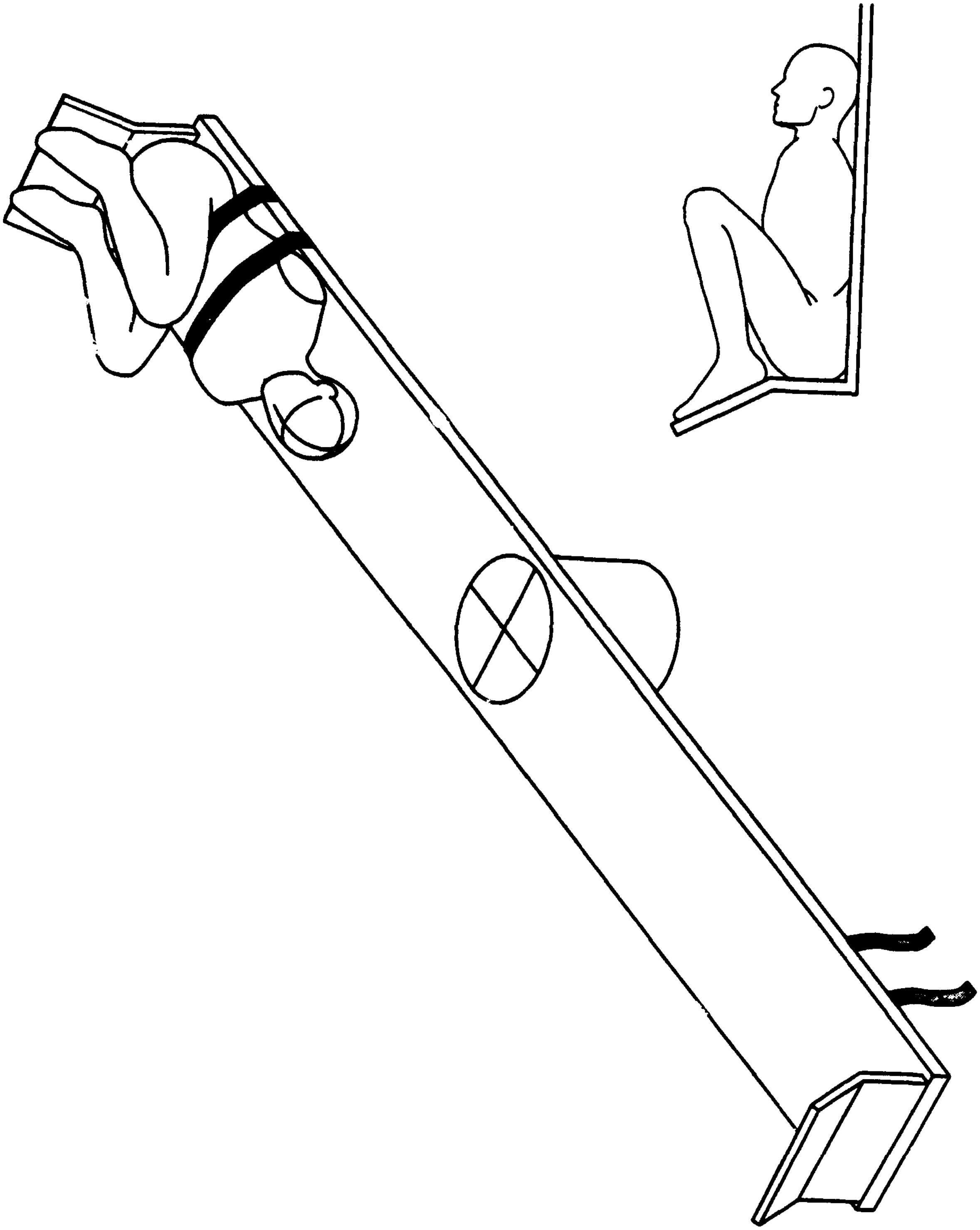


Table 4-1. Crew Anthropometry, Squatting Position

DIMENSION	MALE	FEMALE
Stature		
5th Percentile	169.7 cm	148.9 cm
50th Percentile	179.9 cm	157.0 cm
95th Percentile	190.1 cm	165.1 cm
Head to Heart (z-axis)[1]		
5th Percentile	34.7 cm	29.4 cm
50th Percentile	35.8 cm	30.2 cm
95th Percentile	37.5 cm	30.9 cm
Head to COM (z-axis)[2]		
5th Percentile	51.6 cm	45.2 cm
50th Percentile	53.5 cm	46.7 cm
95th Percentile	55.4 cm	48.0 cm
Head to Buttocks (z-axis)[3]		
5th Percentile	88.9 cm	78.3 cm
50th Percentile	94.2 cm	84.8 cm
95th Percentile	99.5 cm	91.2 cm
Mass		
5th Percentile	65.8 kg	41.0 kg
50th Percentile	82.2 kg	51.5 kg
95th Percentile	98.5 kg	61.7 kg
Moment of Inertia (x-axis)[4]		
5th Percentile	3.8 $kg \cdot m^2$	2.4 $kg \cdot m^2$
50th Percentile	4.9 $kg \cdot m^2$	3.1 $kg \cdot m^2$
95th Percentile	6.0 $kg \cdot m^2$	3.8 $kg \cdot m^2$

[1] Approximate.

[2] Estimated center of mass position for squatting person in microgravity. Female values estimated from ratio of male center of mass position to male stature.

[3] Sitting position.

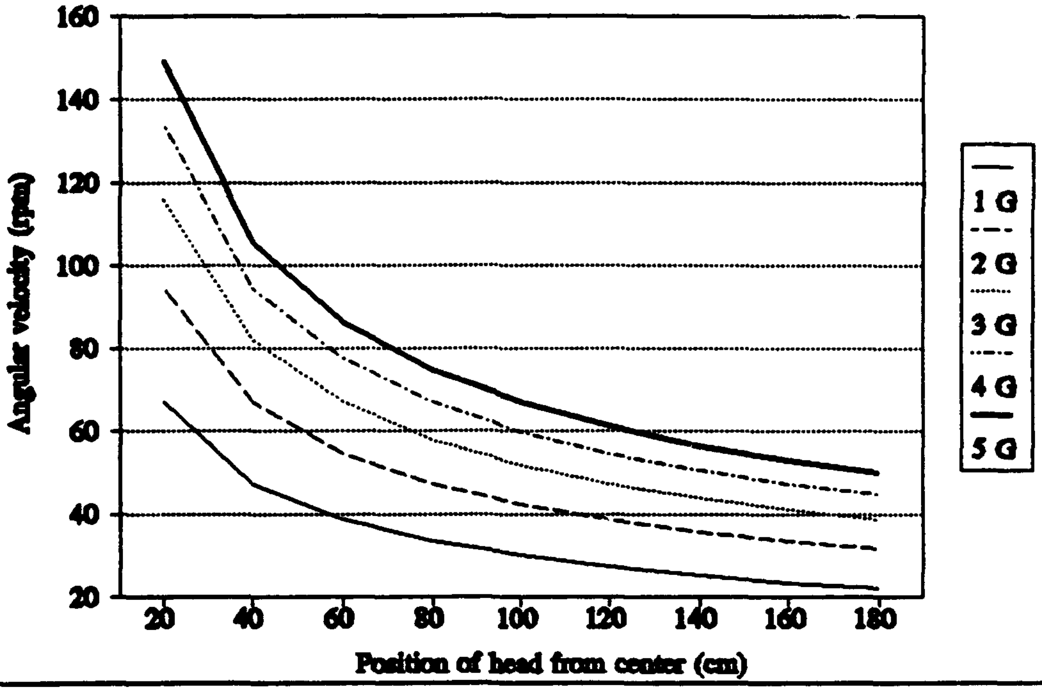
[4] Female values scaled from male values using mass ratio.

Finally, we assume the position of the centrifuge in the shuttle is directly behind the crew compartment, approximately 12.9 m forward of the shuttle center of mass.[1] This value will be used to find the resultant couple moment on the shuttle from reaction forces at the centrifuge axis. Of course, the centrifuge could be located at a different position, with a potentially different effect to the shuttle.

4.3.2 Centrifuge and Shuttle Motion.

From Equation 4-5, radial acceleration of a point on the short arm centrifuge is proportional to the distance of the point from the center and proportional to the square of the angular velocity of the centrifuge. Neglecting the small acceleration component from shuttle motion, Figure 4-3 displays the centrifuge angular velocity required to achieve different radial accelerations as a function of the distance of the crewmember's head from the center of the centrifuge. Typical operation of the centrifuge to achieve useful radial accelerations would require rotation rates between 20 and 80 rpm.

Figure 4-3. Centrifuge Angular Velocity for Different Head Positions



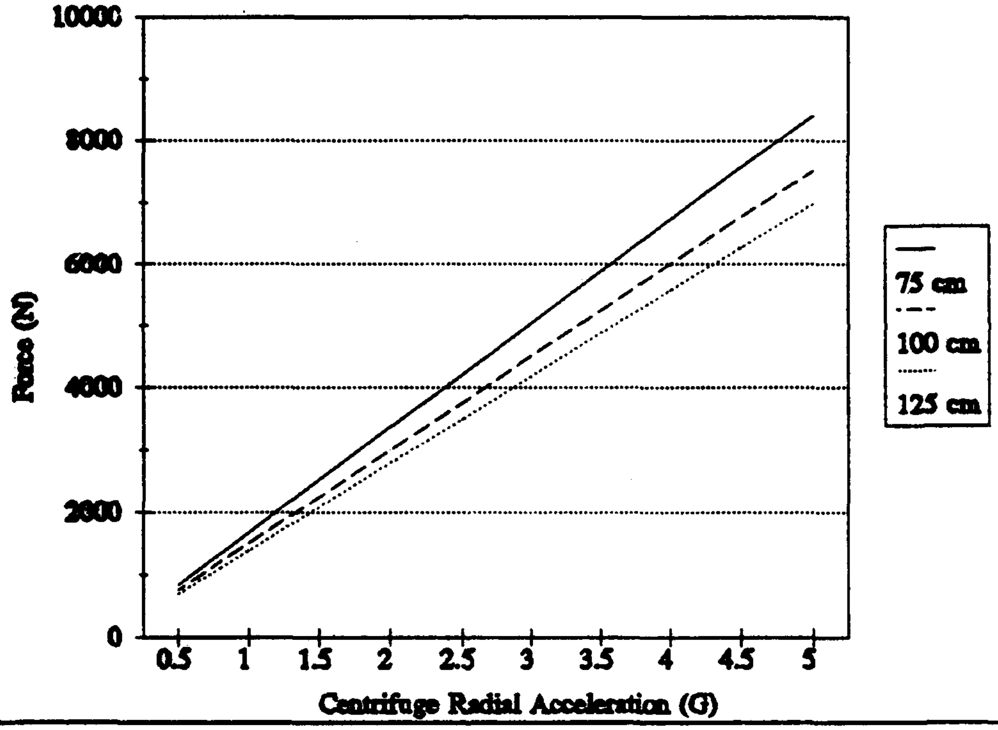
If a single rider is on the centrifuge, the centrifuge would be dynamically unbalanced and would impose a reaction force and moment on the shuttle. Figure 4-4 displays the magnitude of imbalance force as a result of a single 95 percentile male rider with his head at different distances from the center and for different radial accelerations at his head location. Note that in many of the following figures, the radial acceleration at the top of the head and distance from the head to the spin axis will be the abscissa and legend. If the centrifuge is rotating at constant speed, the imbalance force can be represented as a sinusoid with components in the centrifuge y and z axes:

$$F_{imb_Y} = m_p \omega_C^2 r_p \sin(\omega_C t) \ , \qquad [4\text{-}21]$$

and

$$F_{imb_Z} = m_p \omega_C^2 r_p \cos(\omega_C t) \ . \qquad [4\text{-}22]$$

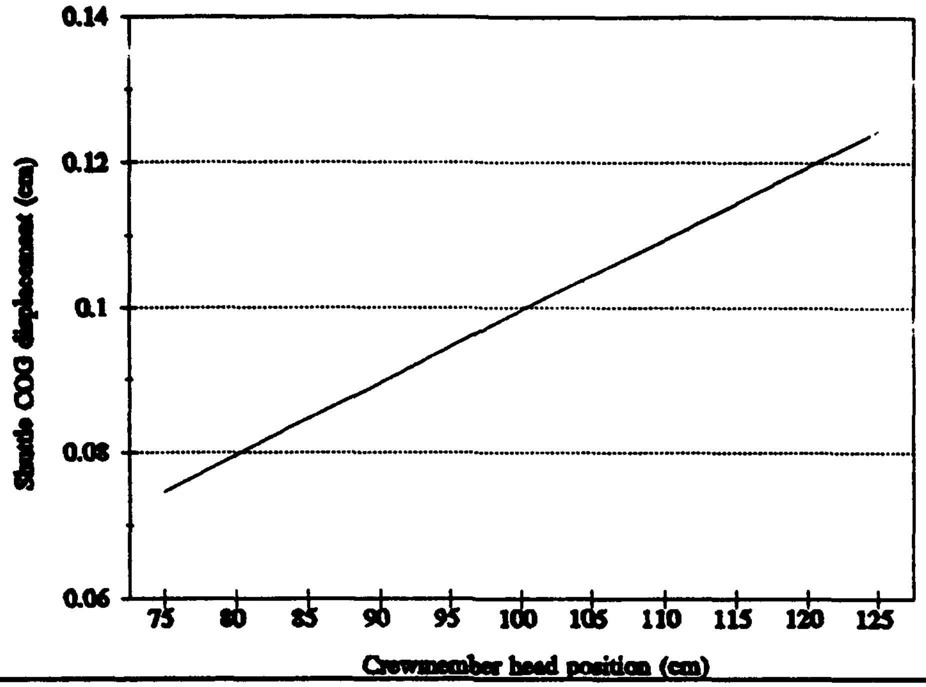
Figure 4-4. Imbalance Force for Different Centrifuge Accelerations and Rider Positions



Since the imbalance force is sinusoidal, displacement of the shuttle center of mass will be in a circular path. Equations 4-21 and 4-22 can be divided by the shuttle mass and integrated twice to yield the linear displacement of the shuttle center of mass. The maximum displacement due to a single 95% rider is on the order of 0.1 cm and is displayed in Figure 4-5. The displacement is computed with the following expression:

$$\Delta r_{SS_{max}} = \frac{|F_{imb}|}{m_{SS}\,\omega_C^2} = \left(\frac{m_P}{m_{SS}}\right) r_P \ . \qquad [4\text{-}23]$$

Figure 4-5. Shuttle C.O.M. Displacement for Different Crewmember Positions



Note from Equation 4-23 that the angular velocity of the centrifuge becomes irrelevant to the steady circular motion of the shuttle.

For the shuttle with mass of approximately 98,000 kg, the centrifuge imbalance force would result in accelerations between 10^{-2} and 10^{-3} G, with a frequency on the order of 1 Hz. At a distance of 12.9 meters forward of the shuttle center of mass, the rider mass imbalance will create a disturbing couple moment. The moment caused by a single 95 percentile male rider at different centrifuge radial accelerations is displayed in Figure 4-6. Obviously, this moment is significant -- on the order of 10^4 - 10^5 N·m. If we assume for simplicity that the shuttle can be represented as a slender rod with uniform mass distribution, displayed in Figure 4-7, the imbalance moment from a single rider would rotate in direction as the person was centrifuged. Recalling Equation 4-10 for angular motion of a rigid body, the angular motion of the rod could be expressed by the following component equations:

Figure 4-6. Disturbance Moment Due to Unbalanced Rider Mass

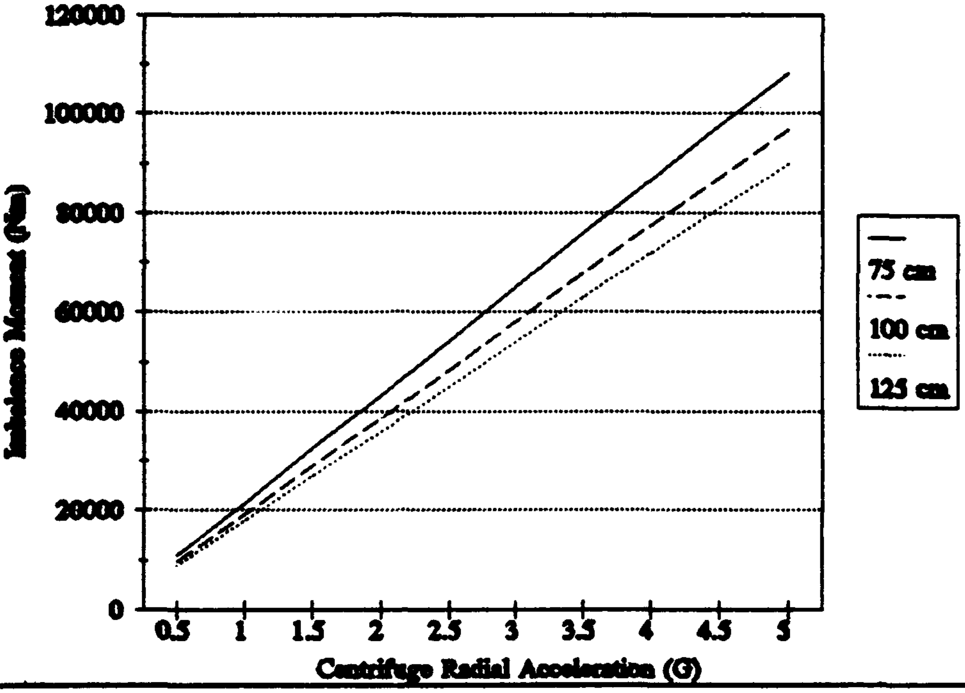
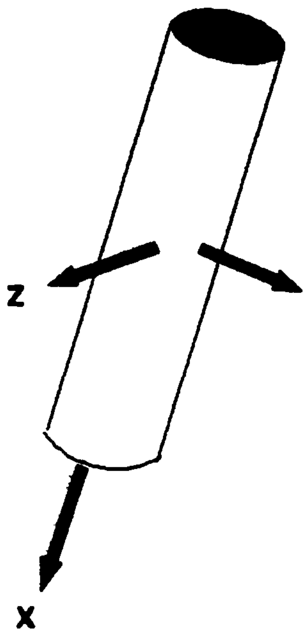


Figure 4-7. Simple Shuttle Model



$$-M_{imb}\sin(\omega_C t) = I_{SS}\alpha_{SS_Y} \ , \qquad [4\text{-}24]$$

and

$$M_{imb}\cos(\omega_C t) = I_{SS}\alpha_{SS_Z} \ . \qquad [4\text{-}25]$$

Integrating twice yields an expression for the approximate maximum angular displacements due to the imbalance moment:

$$\phi_{max_{YZ}} = \frac{M_{imb}}{I_{SS}\omega_C^2} \ . \qquad [4\text{-}26]$$

Substituting a nominal value of 40 rpm for the angular velocity, 10^7 kg·m^2 for the bar inertia, and $5(10)^4$ N·m for the imbalance moment yields a maximum angular displacement of approximately 0.015°. This angular excursion might prove tolerable for some shuttle missions.

The mass imbalance is not the only destabilizing torque created by centrifugation. Angular acceleration of the centrifuge relative to the shuttle to achieve a certain radial acceleration requires a drive moment proportional to the inertia and angular acceleration of the centrifuge. This moment was described in Equation 4-15, and can be alternatively expressed as a function of available power input.[1] Consider that rotational kinetic energy of the centrifuge at a specific angular velocity is defined as:

$$E_{rot} = \frac{1}{2} I_C \omega_C^2 \ . \qquad [4\text{-}27]$$

Knowing that power is the rate of change in energy, Equation 4-27 can be written to yield:

$$\omega_C = \sqrt{\frac{2P_{ave}\Delta t}{I_C}} \ . \qquad [4\text{-}28]$$

Differentiating yields an expression for the angular acceleration of the centrifuge as a function of the available power input and time to accelerate:

$$\alpha_C = \sqrt{\frac{P_{ave}}{2I_C \Delta t}} \ . \qquad [4\text{-}29]$$

Using Equation 4-29, the time to achieve a certain radial acceleration is displayed for different constant average power inputs in Figure 4-8. The average moment transmitted to the shuttle during the centrifuge acceleration is displayed in Figure 4-9. Note that although this moment is several orders of magnitude less than the imbalance moment, it is in a constant direction and is about the axis of smallest shuttle inertia. Furthermore, once the shuttle obtains a constant angular velocity, it will continue to rotate until the centrifuge is slowed. Figure 4-10 displays the angular velocity of the shuttle after the centrifuge accelerates to a certain radial acceleration. Notice that the resultant angular velocity of the shuttle is a function only of the total energy input to the centrifuge. When the centrifuge slows, the shuttle should return to nearly zero angular velocity.

Figure 4-8. Time to Accelerate Centrifuge with Different Power Inputs

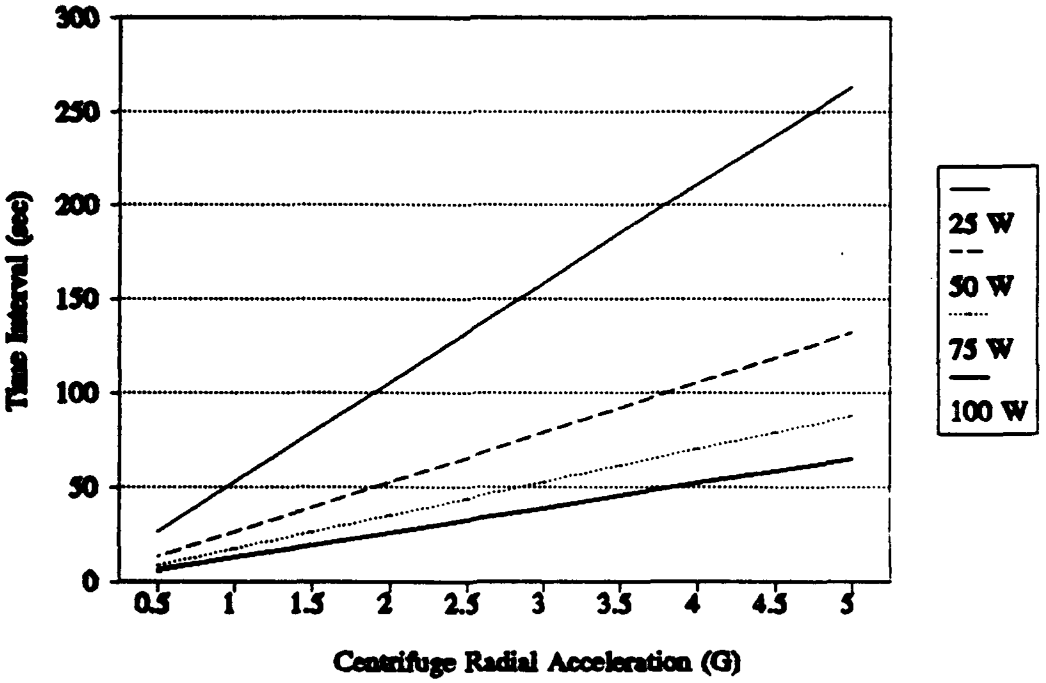


Figure 4-9. Average Centrifuge Drive Moment

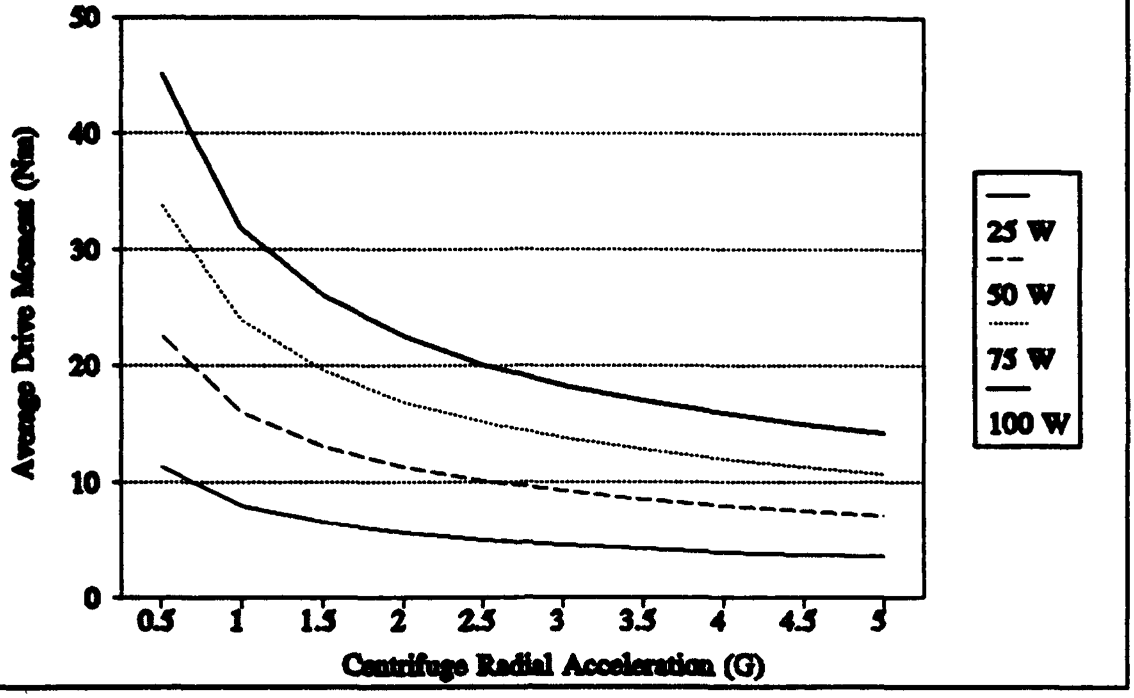
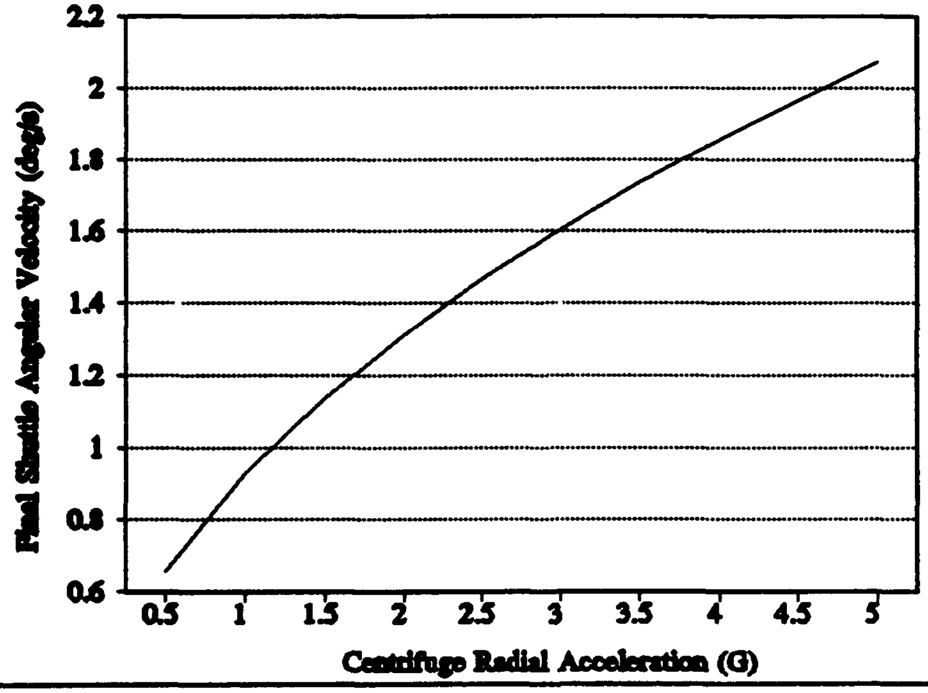
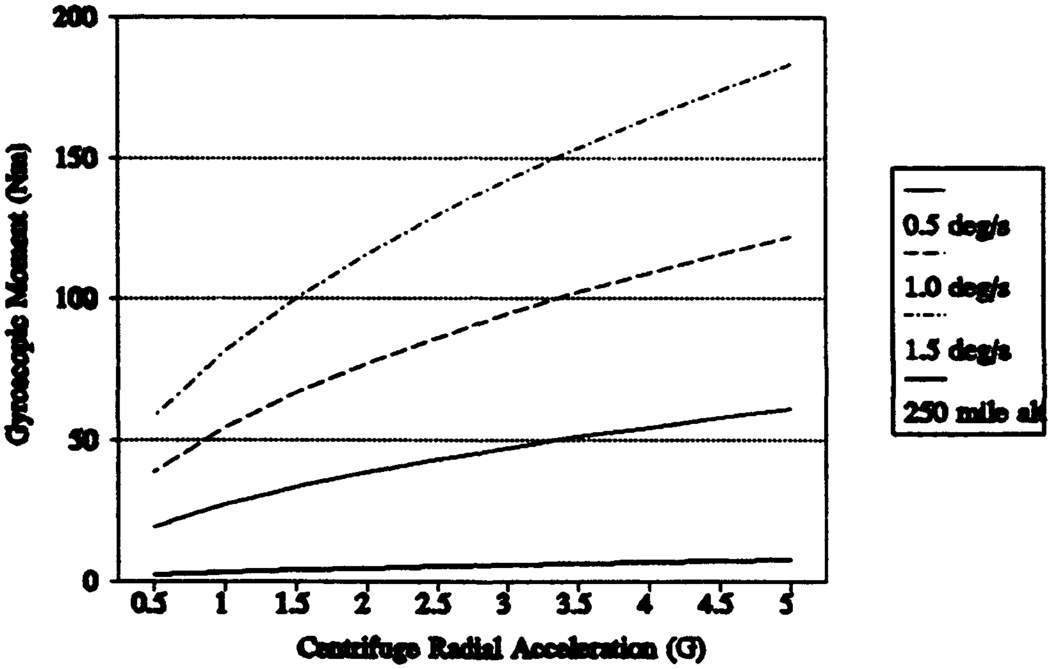


Figure 4-10. Shuttle Angular Velocity from Centrifuge Drive Moment



A third aspect of the centrifuge dynamics could potentially be destabilizing to the shuttle. Attempting to maneuver the shuttle while the centrifuge is in operation will cause a gyroscopic moment to be generated from the changing direction of the centrifuge angular momentum vector. From Equation 4-16, the gyroscopic moment induced by centrifuge rotation for different shuttle angular velocities is plotted in Figure 4-11. There is an additional gyroscopic moment due to the angular velocity of the shuttle around the earth. The gyroscopic moment due to centrifugation at a 250 mile altitude orbit is also plotted in Figure 4-11. Like the drive moment, the gyroscopic moment is much smaller than the imbalance moment and acts about a slowly moving axis.

Figure 4-11. Centrifuge Gyroscopic Moments for Different Radial Accelerations



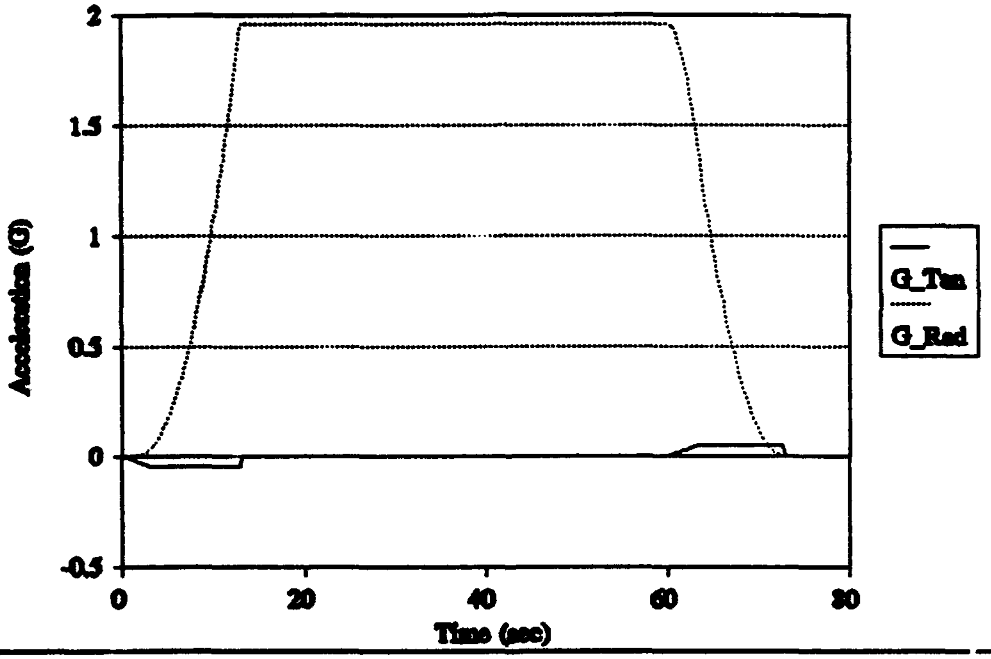
To summarize, there are three potentially destabilizing moments and a single force of concern during centrifugation. Mass imbalance of the centrifuge riders causes both a disturbing force and the largest magnitude moment, although both the force and moment rotate with the centrifuge and have a net effect of zero. The drive moment to accelerate the centrifuge and maintain a constant angular velocity is significant because it is about a constant axis for potentially long durations. Finally, the gyroscopic effect of positioning the shuttle while the centrifuge is in operation creates a relatively small magnitude moment that acts about a slowly moving axis for potentially long durations. In fact, a small gyroscopic moment will exist whenever the centrifuge is operated due to the orbit of the shuttle when the shuttle orientation is maintained with respect to local vertical.

4.3.3 Centrifuge and Shuttle Dynamic Simulation.

A dynamic simulation was conducted of the shuttle during typical centrifuge operation. The simulation featured a single 95 percentile male rider located 100 cm from the center of the centrifuge, with a desired radial acceleration at his head of 2 G. The total time of the simulation was limited to 80 seconds to minimize the amount of data for analysis. There was no attempt to balance the mass of the rider, nor were there any moment compensation devices.

Figure 4-12 displays the tangential and radial acceleration of the centrifuge. After approximately 15 seconds, the rider has achieved a radial acceleration of 2 G and the tangential acceleration is discontinued. At time 60 seconds, the centrifuge is decelerated to a stop.

Figure 4-12. Centrifuge Radial and Tangential Acceleration for Single Rider at 100 cm



As we described in Section 4.3.2, motion of the single rider will cause the shuttle center of mass to translate in a circular path. Figure 4-13 displays the vector components of translation of the shuttle center of mass. The actual data contains smooth sinusoidal motion, however data sampling for plotting creates the slightly jagged look in the figure. During the tangential acceleration period of the centrifuge, the shuttle acquires a small translational velocity that persists after the centrifuge ceases accelerating. It is because of this velocity bias that the displacement of the shuttle center of mass slowly changes from its initial position. The force from the centrifugation is plotted in Figure 4-14.

Figure 4-13. Shuttle COM Displacement from a Single Centrifuge Rider

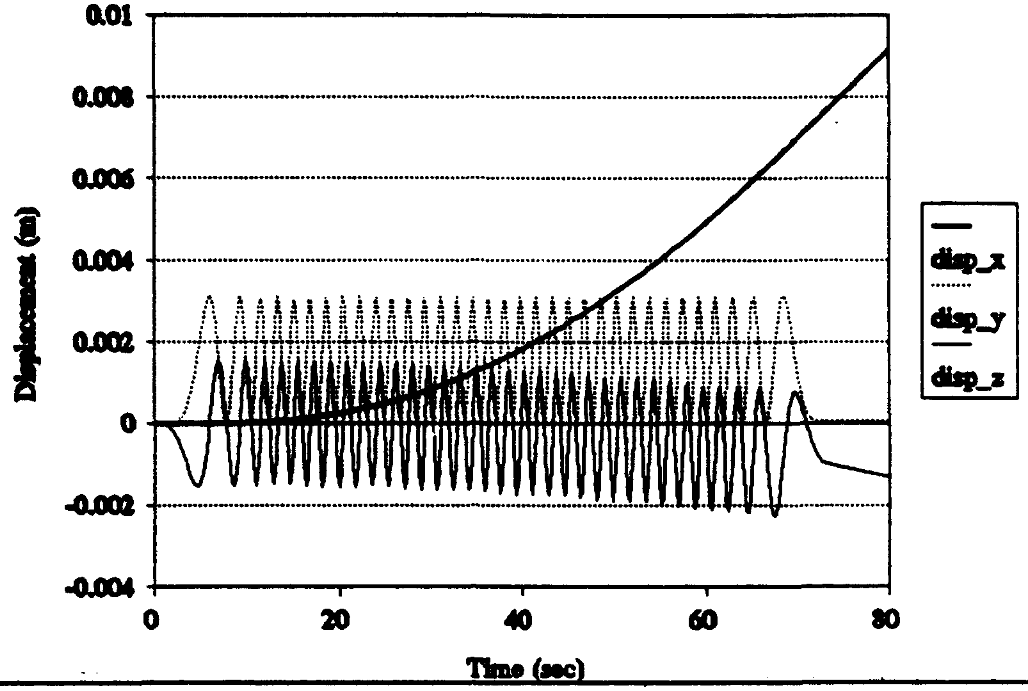
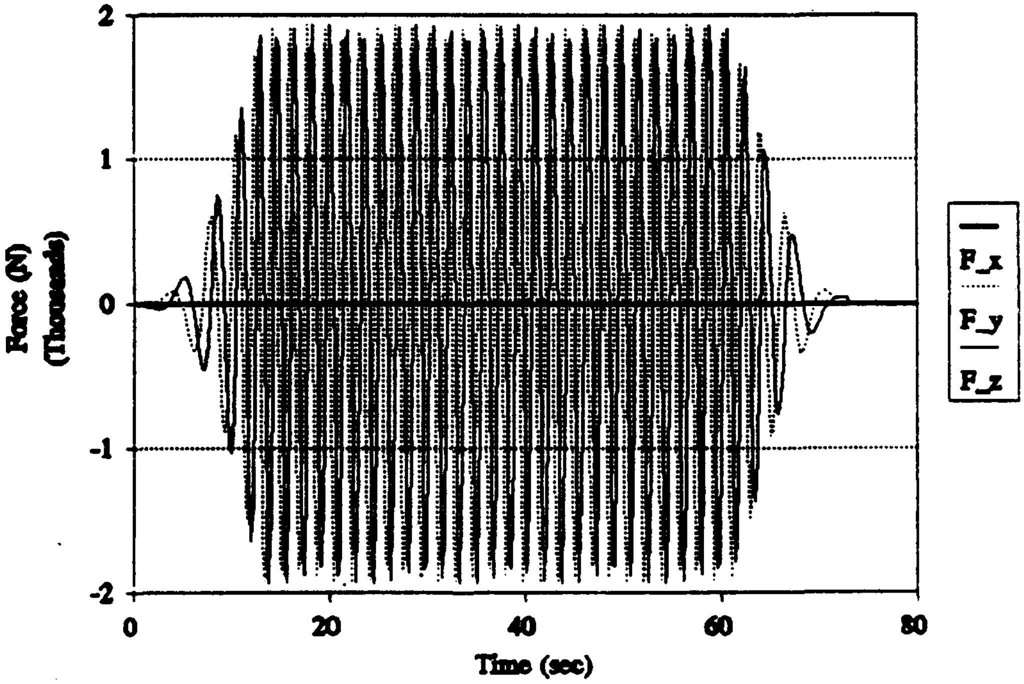
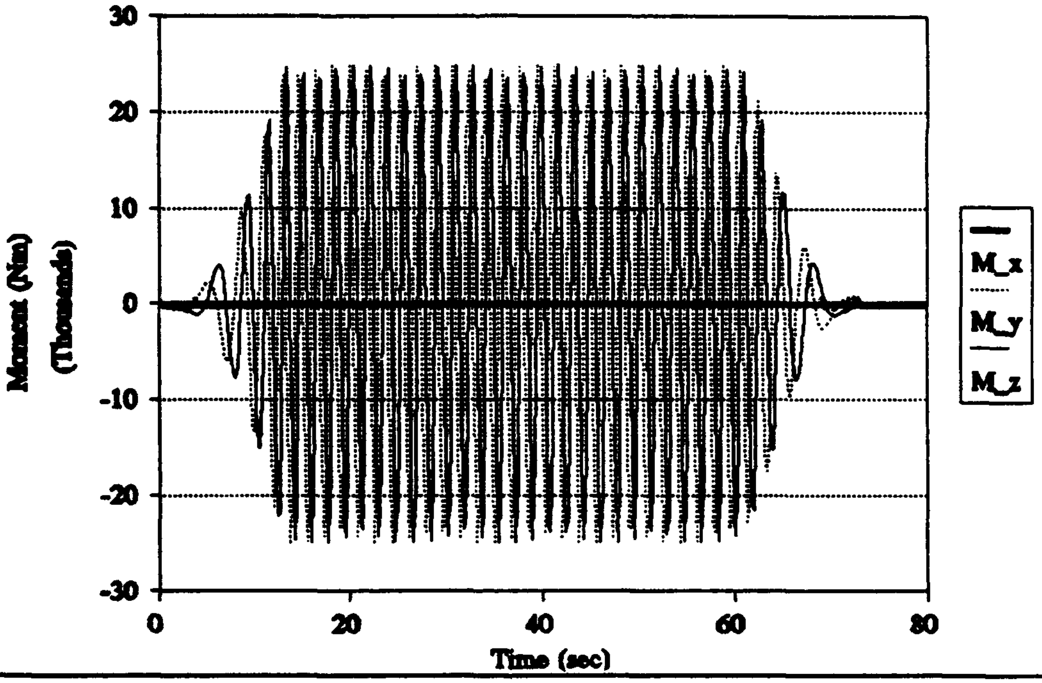


Figure 4-14. Destabilizing Forces to Shuttle from Single Rider Centrifugation



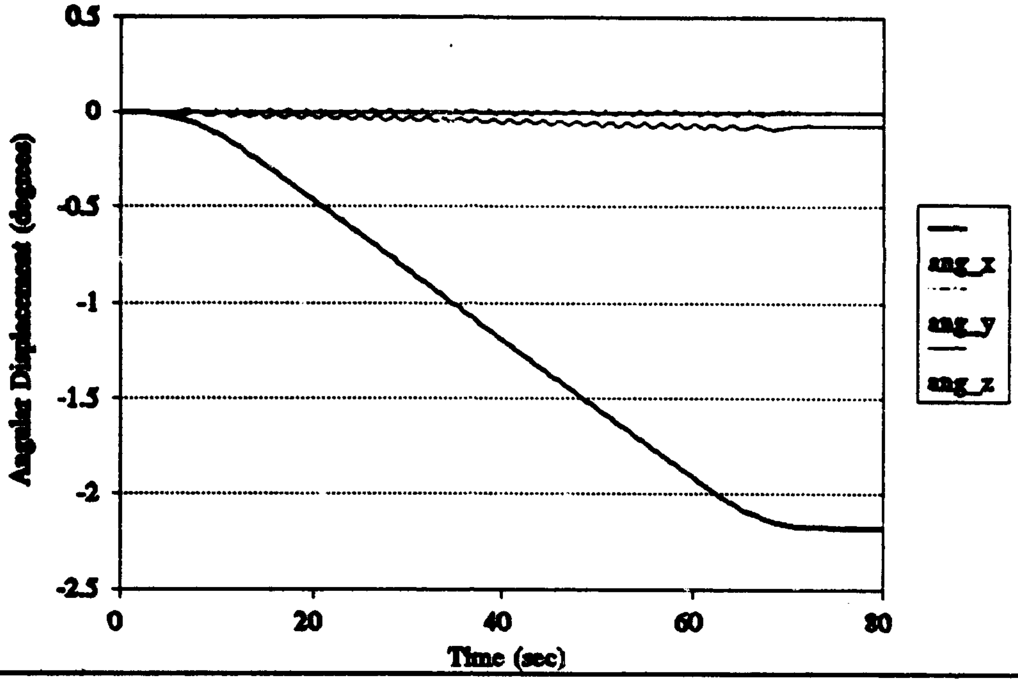
The destabilizing moment transmitted to the shuttle from the centrifuge acceleration and mass imbalance are displayed in Figure 4-15. The small centrifuge drive moment and gyroscopic moment are not noticeable on this figure because of the magnitude of the sinusoidal imbalance moment. As noted earlier, the imbalance moment is on the order of 10^5 N·m.

Figure 4-15. Destabilizing Moments to Shuttle from Single Rider Centrifugation



The effects of the disturbing moments on shuttle orientation are displayed in Figure 4-16. The imbalance moment creates an angular precession of the shuttle in the yz-plane. The centrifuge drive moment generates a constant velocity rotation of the shuttle about the x-axis that continues until the centrifuge decelerates.

Figure 4-16. Shuttle Angular Displacement from Single Rider Centrifugation



4.4 Human Powered Centrifuge.

In Section 3.1, we discussed the use of exercise to aid in preventing Space Adaptation Syndrome and microgravity deconditioning. It is possible that the short arm centrifuge could be powered by riders using a pedalling mechanism to drive the centrifuge.[1,12] While this provides the dual benefit of exercise and acceleration, it also requires additional centrifuge complexity and introduces a new potential source for destabilizing moments to the shuttle. A functional sketch of how the pedals might power the centrifuge is displayed in Figure 4-17.

An average person can exert between 25 and 100 Watts of power during moderate levels of exercise.[13,14] Typical pedalling speeds for cyclists range from 40 to 80 rpm, and pedal torques range from 20 - 60 N·m during cycling.[13,14] Thus, similar to an automobile engine, the pedalling of a centrifuge rider will be limited by the torque he can generate at low rpm and by the power output he can maintain at high rpm. Figure 4-18 displays the torque required for pedalling at different speeds at different average power outputs.

Figure 4-17. Centrifuge with Pedalling Rider

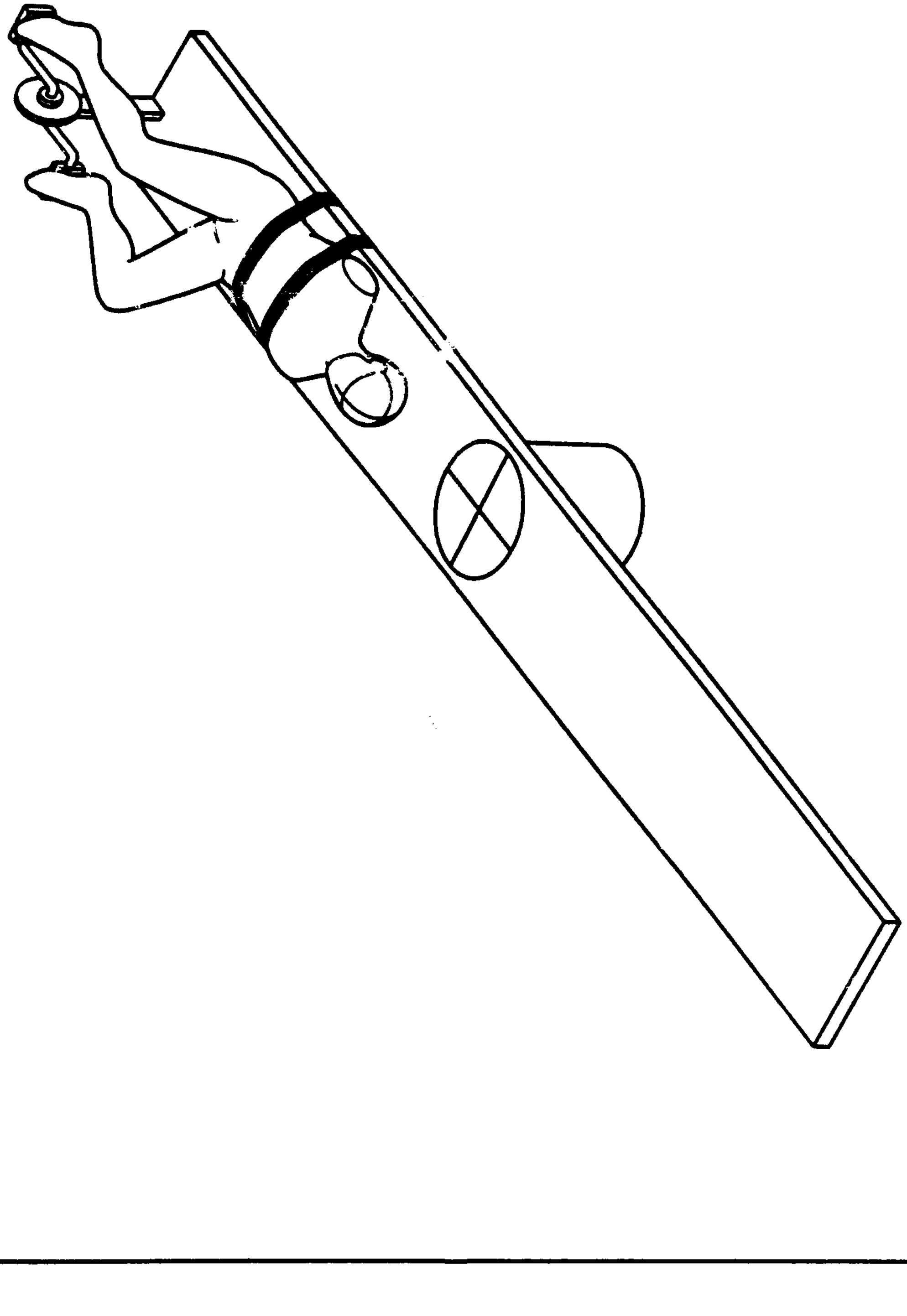
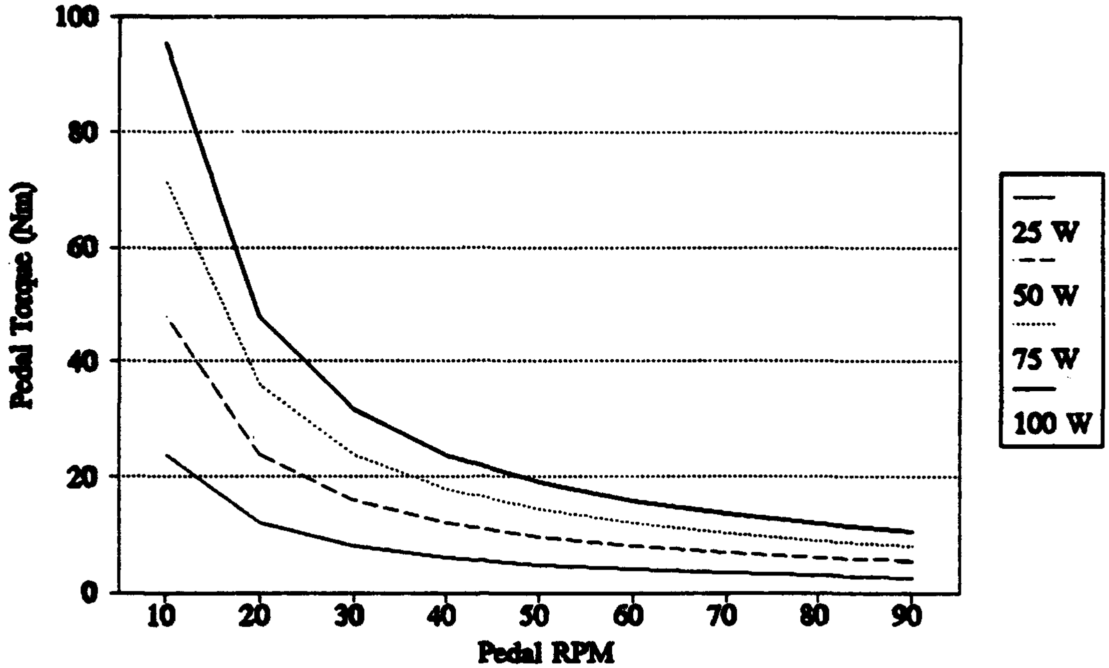


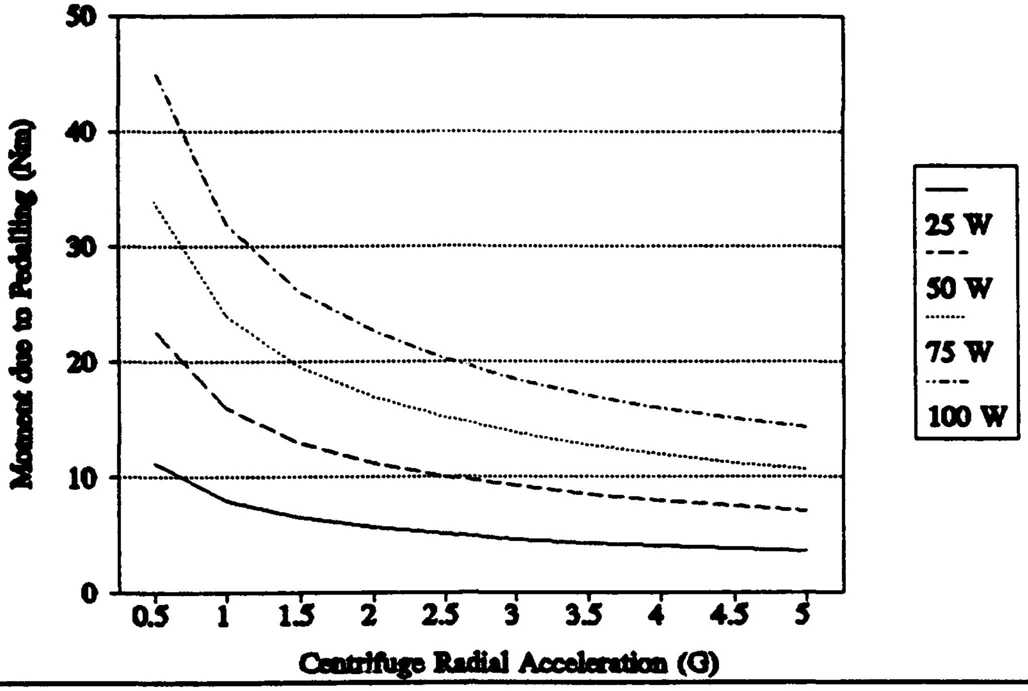
Figure 4-18. Torque-Speed Curve for Pedalling



While the centrifuge is accelerating, a drive moment equal to the product of the centrifuge inertia and angular acceleration will be transmitted to the shuttle, as we described in Equation 4-16. Pedalling power from the rider will be used to create the drive moment through either a mechanical or electrical power transfer. The moment required to rotate the pedals can be less than, equal, or greater than the centrifuge drive moment by using different effective gear ratios in the power transfer. Since the torque and rpm range of Figure 4-18 is similar to that required for the centrifuge (see Figures 4-3 and 4-9), we will assume there is a net gear ratio from pedals to the centrifuge of unity.

Once the centrifuge reaches the desired speed, the pedalling torque should drop to a level required to negate aerodynamic and friction effects. Since the purpose of the pedals is to permit extended periods of exercise, a source of drag would have to be provided. A variable friction load applied near the location of the pedals results in a constant moment being transmitted to the shuttle that rotates in the centrifuge yz-plane as the centrifuge spins. This moment will have a similar effect on the shuttle to the imbalance moment described earlier, except that it will create a much smaller angular excursion of the shuttle. The moment transmitted to the shuttle, assuming the rider is located 100 cm from the spin axis and there is a unity gear ratio from pedals to the centrifuge, is displayed for different radial accelerations and exercise power outputs in Figure 4-19.

Figure 4-19. Disturbance Moment to Shuttle from Subject Pedalling

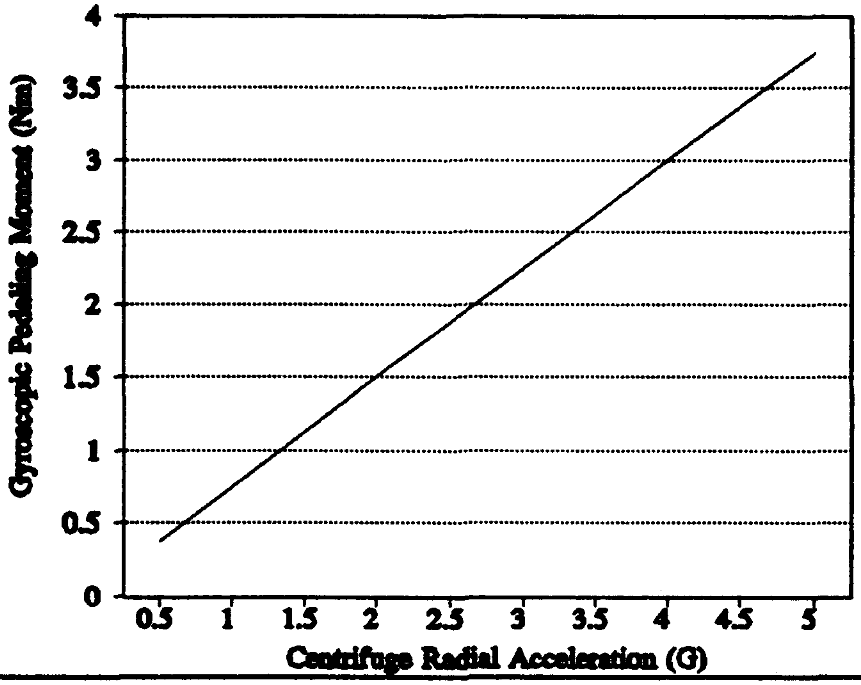


Another potentially destabilizing moment created by the motion of the pedals is a gyroscopic moment that is proportional to the inertia of the pedals and rider's feet, the angular velocity of the pedals, and the angular velocity of the centrifuge:

$$M_{ped_{gyro}} = \omega_C \times I_{ped}\omega_{ped} \ . \qquad [4\text{-}30]$$

Unlike the gyroscopic moment caused by centrifuge and shuttle motion, which was present only when the shuttle was repositioning (recall Equation 4-16), the gyroscopic moment due to pedalling motion is always present while the centrifuge is spinning. This moment is quantified in Figure 4-20 for different centrifuge radial accelerations. From the figure, it evident that this gyroscopic moment is nearly as large as some of the other destabilizing moments, and changes direction as the centrifuge rotates. Approximations of the moment of inertia of the rider's feet are presented in Appendix A1. Using the procedure described in Section 4.3.2, where the shuttle was modelled as a slender bar, the gyroscopic moment caused by pedalling would produce maximum shuttle deviations on the order of approximately 10^{-5} degrees. It is important to note that the moment transmitted to the shuttle by pedalling and the gyroscopic moment created by rotating the feet will always act in the shuttle y-z plane, but be out of phase by 90 degrees.

Figure 4-20. Gyroscopic Moment Due to Pedalling



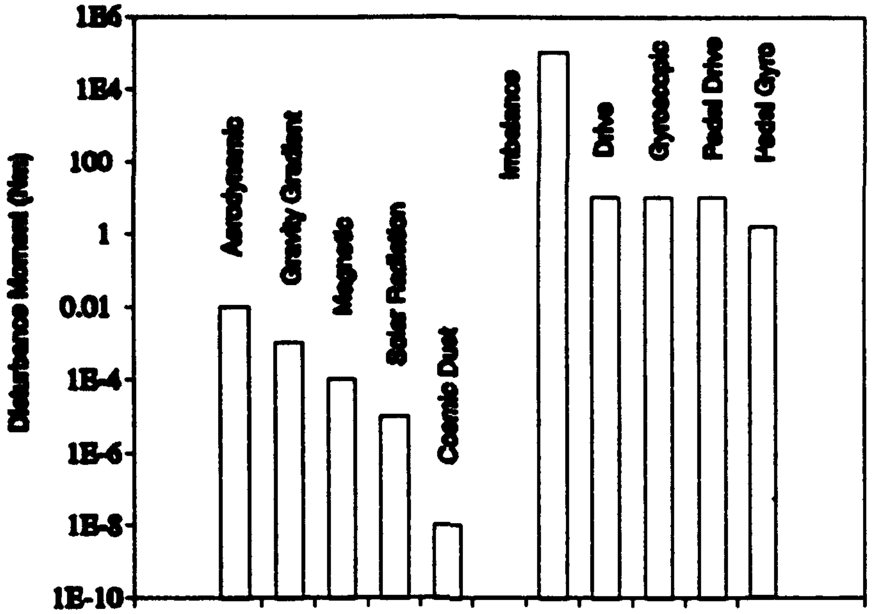
4.5 Dynamic Balancing and Moment Compensation.

To understand the significance of the destabilizing moments created by centrifuge operation and pedal power, the orders of magnitude of the different moments are displayed in Figure 4-21 along with typical environmental disturbance torques.[15]

The reaction moments from centrifuge operation are obviously orders of magnitudes greater than the minute moments associated with environmental disturbances. It is our goal in this section to discuss methods for reducing the destabilizing moments of the centrifuge to a level, if possible, that approaches environmental disturbances.

Of course, one item that has been neglected is the existing space shuttle momentum management system. The shuttle is equipped with various devices to assist in the maintenance of attitude under disturbances from environmental and equipment sources and astronaut motion. Significant moments applied for a relatively long duration can saturate the momentum management system, requiring positioning thrusters to be used. Since the shuttle carries a limited supply of fuel, it is important to avoid extended duration moments.

Figure 4-21. Comparison of Centrifuge and Environmental Disturbance Moments



4.5.1 Dynamic Balancing.

From Figure 4-21, the force created by the dynamic imbalance of the centrifuge with a single rider generates the largest disturbance moment. As we described in Section 4.3.2, the imbalance force creates accelerations between 10^{-2} G and 10^{-3} G of the shuttle with a frequency on the order of 1 Hz. Although the magnitude of the force seems high, studies of an animal centrifuge for use on the Space Station Freedom estimate nominal crew movements and space station operations such as docking, pointing, and various fans, blowers, etc., would impart accelerations on the order of 10^{-3} G, with a frequency range of 10^{-3} to 10 Hz.[16,17,18] These ranges indicate the force and frequency of an unbalanced centrifuge may be acceptable for the shuttle. Furthermore, if the imbalance force is deemed acceptable, presumably the moment caused by the force acting at a distance from the shuttle center of mass with the same frequency would also be tolerable. On the other hand, the centrifuge would find its greatest utility in a relatively frequent operation on very long duration missions for which conservation of thruster fuel would assume greater importance.

Since it is not known if the mass imbalance on the centrifuge will cause unacceptable deviations in attitude of the shuttle, methods for reducing the imbalance force, and hence imbalance moment, are considered in the following subsections.

4.5.1.1 Opposite Riders.

The simplest way to reduce the imbalance is to position a second rider on the opposite side of the centrifuge. If the subject had equal mass and was equidistant from the center, the centrifuge would be perfectly balanced. More likely, the mass of the second subject will be different from the first and the position, although it can be roughly predicted, will vary slightly each time the subjects are situated. To quantify the effects of partial imbalancing, consider Figure 4-22. This figure displays the residual imbalance moment for different radial accelerations after placing a 5 percentile female rider equidistant but opposite a 95 percentile male rider.

Although an improvement of 50% or more, the imbalance is still significant. Even if the masses of the two riders are equal, the effects of position are significant. Figure 4-23 shows the resulting imbalance moment for different accelerations resulting from two 95 percentile male riders being positioned at approximately 100 cm from the center, both with some differential position. It is conceivable that subjects could easily have a 5 - 10 cm position error, which still results in a significant imbalance moment.

Figure 4-22. Centrifuge Imbalance Moment with 5% and 95% Riders

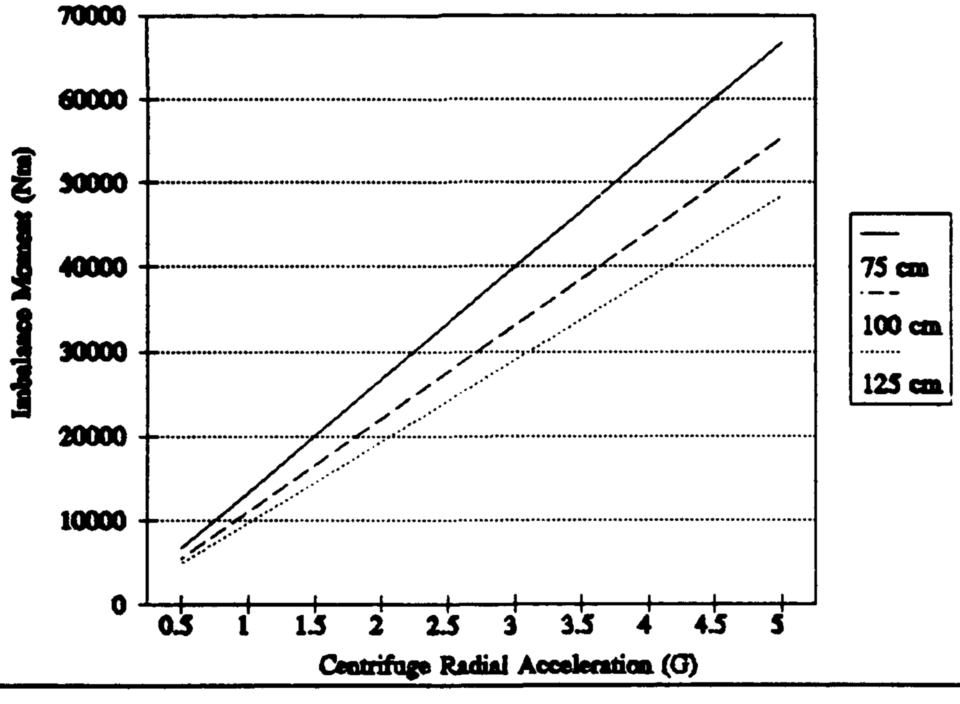
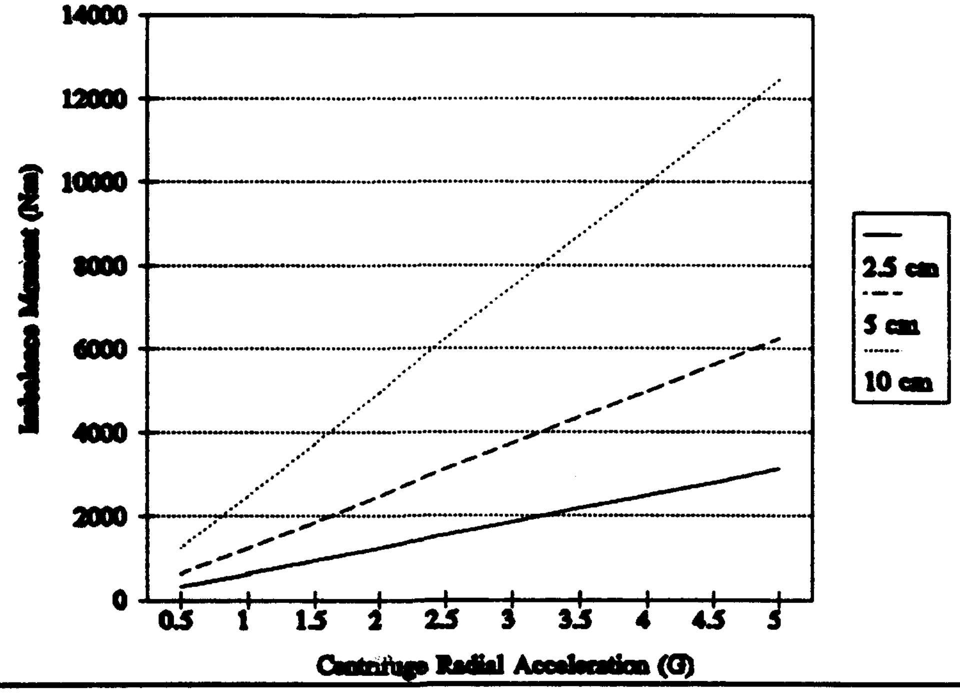


Figure 4-23. Centrifuge Imbalance Moment with 2 Riders at Different Positions



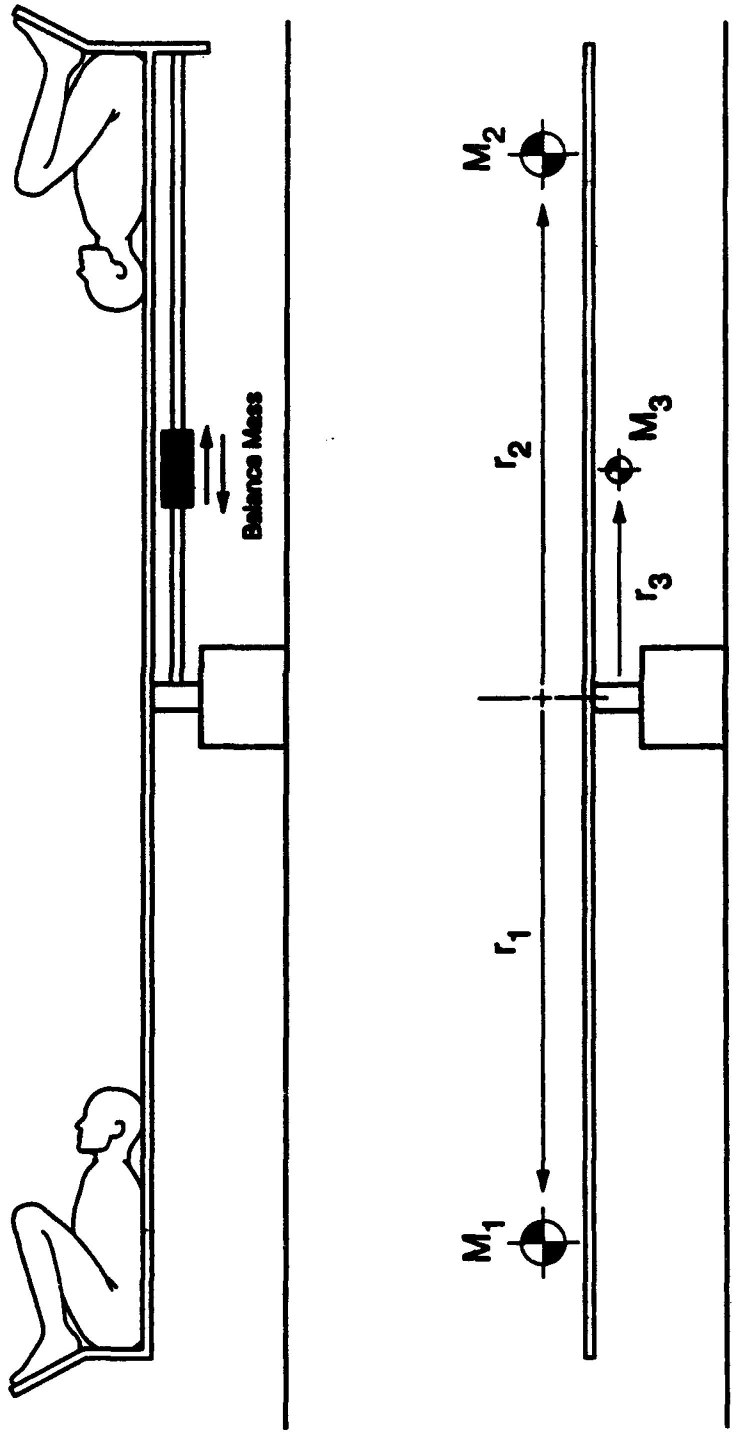
4.5.1.2 Counterbalance Mass.

Similar to the concept of an opposing rider, a large counterbalance mass could be used to balance the centrifuge. The counterbalance mass, hereafter called countermass, could probably be more reliably positioned than a human, however the mass would be significant and undesirable. As an alternative, consider a combination of opposing rider and small countermass. By positioning the large and small rider equidistant from the centrifuge center, the countermass can be used on the side of the small rider to move the combined center of mass back to the center of rotation.

It is important to quantify how much mass and how much motion is required for the countermass to be effective. Figure 4-24 displays a schematic of the centrifuge with two riders and moveable countermass. Dynamic balance can be achieved by locating the countermass, m_3, such that the center of mass of the system is at the center of the centrifuge, or

$$m_1 r_1 - m_2 r_2 - m_3 r_3 = 0 \quad . \qquad [4\text{-}31]$$

Figure 4-24. Countermass Concept for Two-Rider Centrifuge

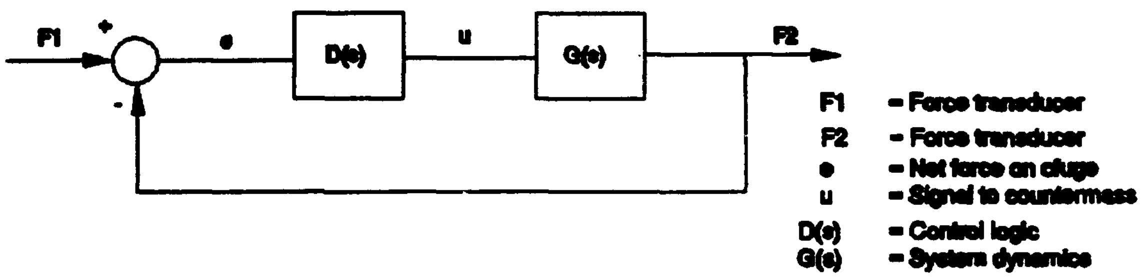


Note that it is not necessary for the sum of the small rider's mass and countermass to equal the mass of the large rider. Because of this, the size of the countermass can be minimized by allowing it to travel to the edge of the centrifuge, approximately 200 cm from the center.

For the worst case of a 95 percentile male and 5 percentile female, each located with their heads 100 cm from the center, a countermass of 46.8 kg would be required at 200 cm on the side of the female. For the case of two equal mass subjects positioned equidistant from the center, the same 46.8 kg countermass could be moved to the center of the centrifuge and have no effect on the center of mass. Since the countermass is fairly large, it might also be desirable to position the female further from the center than the male. In the case of equidistant subjects, however, the ideal countermass to accommodate the maximum and minimum subjects is 46.8 kg, with a range of travel from the center to a 200 cm radius. If desired, the countermass could simply be a shell, to be ballasted with available materials in space. In this way, the liftoff mass would be minimized.

To make the balancing automatic, we propose the motion of the countermass be controlled by a closed-loop controller with load cell feedback from the center of centrifuge. Figure 4-25 displays a block diagram of a conceptual balance controller. Two load cells, on opposing arms of the centrifuge, measure the force F_1 caused by radial acceleration of the heavy subject and F_2 caused by radial acceleration of the lighter subject and countermass. The control algorithm, represented by D(s), would generate a signal, *u*, to move the countermass until the difference between the two load cell values, *e*, is zero. The actuator for moving the countermass is not specified here, however one solution might be to mount the countermass on railings and use tension in wire attached to both ends to pull the countermass in the required direction.

Figure 4-25. Mass Balancing Controller Block Diagram



Depending on the design of the positioning mechanism, it might be possible to move the countermass fast enough to balance transient motion of the riders, such as movement of arm or leg position. Even if the response of the balancing system was too slow for transient center of mass deviations, the system could adapt to a steady state change in system center of mass, such as a rider moving his hands from his sides to over his chest, perhaps.

Certainly many details of a dynamic balancing mechanism and controller require further consideration. There is little discussion of the positioning mechanism presented here. In spite of these details, however, it is likely an effective balancing mechanism could be developed that would reduce the imbalance moment to a tolerable value.

4.5.2 Moment Compensation.

Even if the imbalance moment can be nearly eliminated, significant moments are created by the centrifuge drive, cross-coupling of angular velocities, and subject pedalling. In the following subsections, compensation for these moments is discussed.

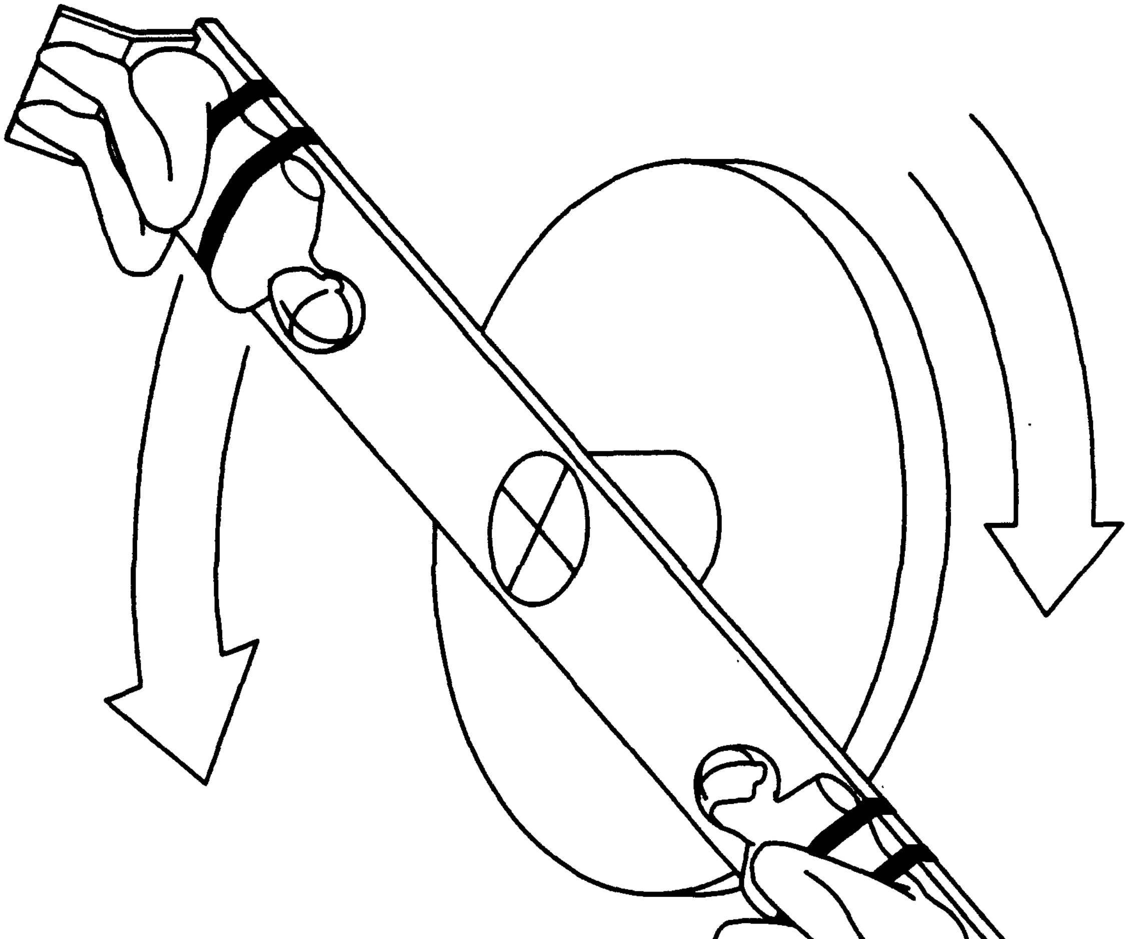
4.5.2.1 Counterbalance Inertia.

The space shuttle may already be equipped with a reaction wheel or other device to prevent rolling about the shuttle x-axis. If this is the case, it may not be necessary to eliminate the centrifuge drive moment. Neglecting any shuttle compensation devices, a simple method to eliminate the drive moment is by angular acceleration of an object in the opposite direction, as shown in the sketch in Figure 4-26.

Rotation of a counterbalance inertia, shortened to counterinertia, in the opposite direction of the centrifuge will create a moment of the opposite sign. Since a moment applied to a rigid body has the same effect regardless of the point of application, it is not necessary to rotate the countermass at the centrifuge location, although it may be preferable for other reasons. Furthermore, the counterinertia need not have an identical inertia to the centrifuge and riders. A smaller inertia can be accelerated at a faster angular acceleration over the same duration of time to compensate for the drive moment. From shuttle mass considerations, a counterinertia of low mass and low inertia, driven at a high angular acceleration, is the most desirable.

The same counterinertia that negates the drive torque can be used to reduce or eliminate gyroscopic moments due to shuttle positioning and shuttle motion about the earth. If the centrifuge and counterinertia have the same angular momentum, the centrifuge-induced gyroscopic moment is eliminated. It is important to determine whether a low inertia counterinertia propelled at high angular accelerations to eliminate the drive torque will result in an equal and opposite angular momentum to the centrifuge, thereby also eliminating the centrifuge gyroscopic moment. Consider the torque required to angularly accelerate the centrifuge, described previously in Equation 4-15, and expressed as a scalar:

Figure 4-26. Counterinertia Concept for Two-Rider Centrifuge



$$M_{drv_C} = I_C \alpha_{C_{SS}} \; . \qquad [4\text{-}32]$$

If the counterinertia, denoted by the subscript CI, is allowed to accelerate at a different rate than the centrifuge, the moment of inertia of the counterinertia required to eliminate the drive torque is:

$$I_{CI} = I_C \left(\frac{\alpha_C}{\alpha_{CI}} \right) \; . \qquad [4\text{-}33]$$

If the centrifuge is propelled with a constant angular acceleration for a time t, the angular momentum of the centrifuge at the end of this time will be:

$$H_C = I_C \omega_C = I_C \alpha_C t \; . \qquad [4\text{-}34]$$

Similarly, the angular momentum of the counterinertia after the same time interval is:

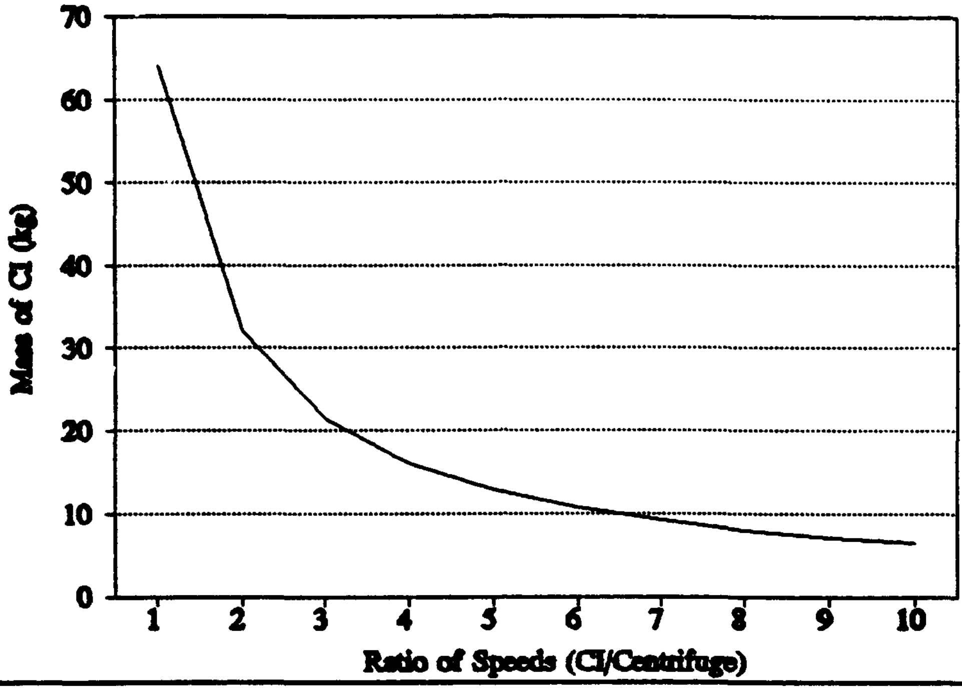
$$H_{CI} = I_{CI} \omega_{CI} = I_{CI} \alpha_{CI} t \; . \qquad [4\text{-}35]$$

Recalling the constraint of Equation 4-33, which defines the inertia of the counterinertia required to eliminate the drive torque, it can be seen that substitution of Equation 4-33 into 4-35 yields a counterinertia angular momentum that is equal and opposite to the angular momentum of the centrifuge. Therefore, it is possible to use a counterinertia mechanism with lower mass and inertia and higher angular acceleration and velocity than the centrifuge and still eliminate both the centrifuge drive and gyroscopic moments.

One way that the counterinertia could be implemented is to use a variable speed mechanical or electrical drive and fixed inertia. If the speed was variable over a large range, any magnitude torque could be generated. The difficulty in this concept, however, is the use of a variable speed drive. Mechanical and electrical variable speed drives might add too much complexity and mass to be considered.

An alternative counterinertia design would feature a variable inertia mechanism and fixed angular acceleration ratio relative to the centrifuge. The counterinertia drive could be provided by a mechanical connection with fixed gear ratio, or by a separate electrical drive. To provide a variable inertia, it is envisioned that two masses could be mounted on rails and positioned outwards in opposite directions. Similar to the dynamic balancing mechanism, a closed-loop control system could determine the optimum position for the masses to negate the centrifuge drive moment. Assuming the worst case of two 95 percentile male riders positioned 100 cm from the center and a centrifuge inertia of 25 kg·m^2, the adjustable mass on each side to eliminate the drive moment is plotted in Figure 4-27 as a function of the ratio of counterinertia angular velocity to centrifuge angular velocity.

Figure 4-27. Mass of Counterinertia for Different Angular Velocity Ratios



4.5.2.2 Opposite Centrifuges.

Another method for reducing the drive moment is by propelling a second two-rider centrifuge in the opposite direction of the original centrifuge. Because of the microgravity environment, it would not matter physiologically that the subjects are facing opposite directions. If the inertias were identical and both centrifuges were accelerated at the same rate, the drive moments would be equal and opposite. More likely, the inertias would not be identical, leading to a situation analogous to that described in Section 4.5.1.1, where an additional mass was required to balance riders of different size. In this case, a counterinertia device could be used to balance any residual drive moment from the two centrifuges.

While opposing two-man centrifuges are inviting in order to reduce the mass of the counterinertia mechanism, there are several considerations that may make the concept less appealing than the single centrifuge with counterinertia concept. For instance, the requirement for four subjects to be centrifuged simultaneously, and probably a fifth to monitor the training, implies that a majority of the shuttle crew is required whenever the device is operated. In addition, the power to drive the centrifuge is increased because of the additional inertia.

4.5.2.3 Pedalling Compensation.

As described in Section 4.4, the exercise load for the pedals would be produced along the pedal axis. The moment transmitted to the shuttle would rotate in the centrifuge yz-plane similar to the mass imbalance moment described in Section 4.3.2. Substituting pedal moment values into Equation 4-23 yields an angular excursion of the shuttle on the order of 10^{-4} degrees. It is possible a moment that causes a misorientation of this magnitude does not require further consideration.

The gyroscopic moment created by the rotating pedals and subject's feet also produces a small precession of the shuttle in the shuttle yz-plane. A properly sized counterinertia could be mounted at the pedals and driven in the opposite direction of pedalling to reduce the transmitted moment. It is unlikely the inertia could be perfectly sized, but a significant portion of the transmitted moment could be eliminated.

The use of a two-rider centrifuge would also tend to compensate for both disturbing moments from pedalling. The compensation would only be ideal if the two riders had equivalent foot and leg characteristics, the pedal axes were equidistant from the centrifuge center, and the pedalling forces for the two riders were equally shared.

4.6 Centrifuge Concept Design.

A design concept for the space-based centrifuge is displayed in Figure 4-28. The centrifuge has accommodations for two crewmembers, and at least one other crewmember would be required to monitor the operation. The centrifuge will be propelled by the combined power of both crewmembers pedalling. An automated mass balancing system and moment compensation devices minimize the disturbances to the orbiting shuttle.

The destabilizing forces and moments before balancing and compensation are identified in Figure 4-29. The direction of the force or moment is indicated by an arrow. Several of the forces and moments displayed are shown pointing in an instantaneous direction, determined by the position of the centrifuge. The imbalance force and moment, pedalling moment, and pedal gyroscopic moment all rotate as the centrifuge spins, so that the net effect if they are of constant magnitude is to induce a circular translational motion at the shuttle center of mass and a steady precession. The centrifuge drive moment and gyroscopic moment, conversely, act on fixed or slowly moving axes. The gyroscopic moment points orthogonally to the centrifuge spin axis and the axis of shuttle angular motion. The centrifuge drive moment may be the most significant, because it induces a constant roll angular velocity in the shuttle that lasts until the centrifugation is ceased.

Mass balancing of the centrifuge could be accomplished with a 47 kg counterbalance mass. This could be a solid object, or to save liftoff mass, it might be a shell filled with a fluid or small objects during orbit. To use the centrifuge with only a single crewmember, some shuttle gear would need to be strapped in place of the missing rider.

Figure 4-28. Centrifuge with Two Riders

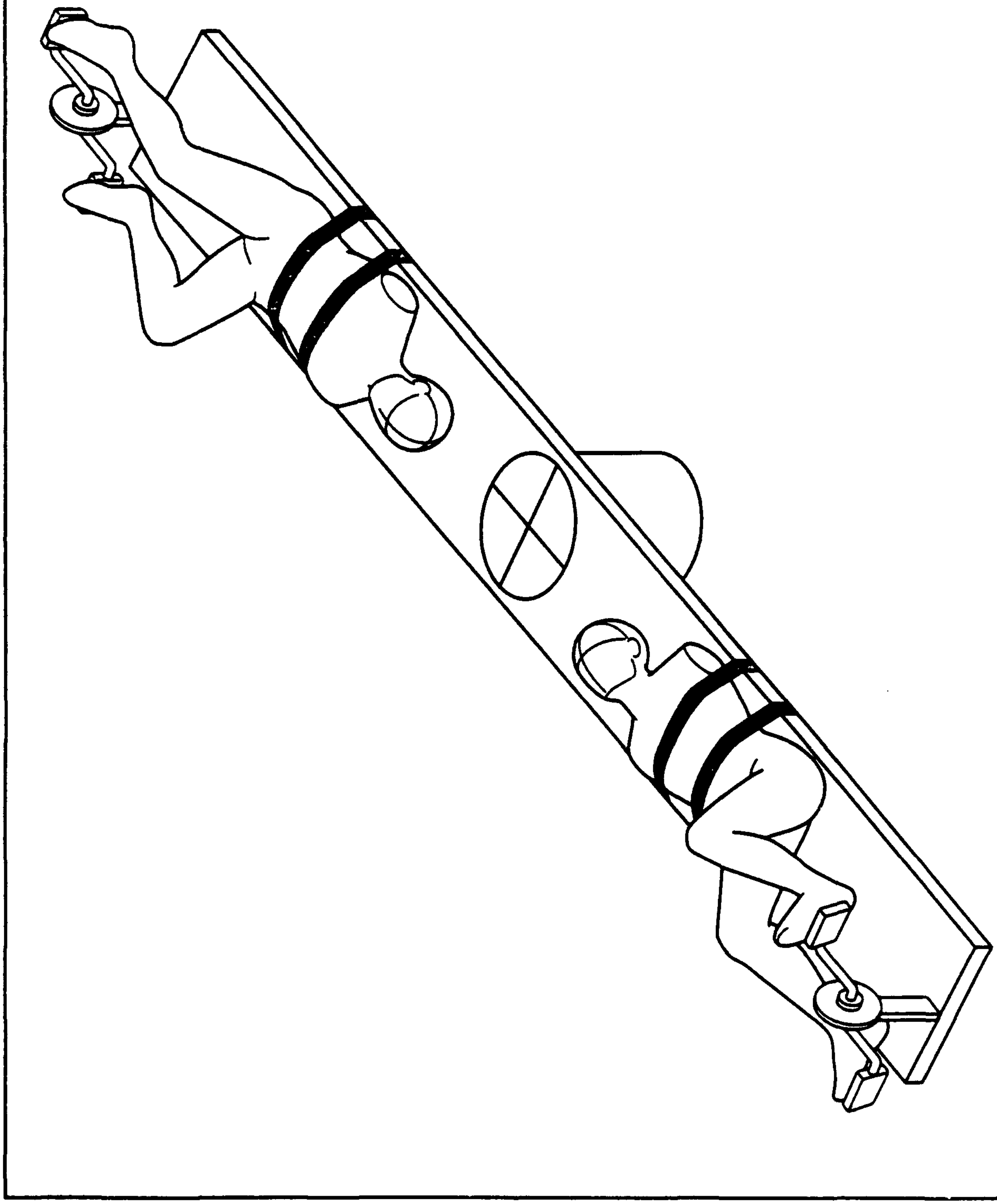
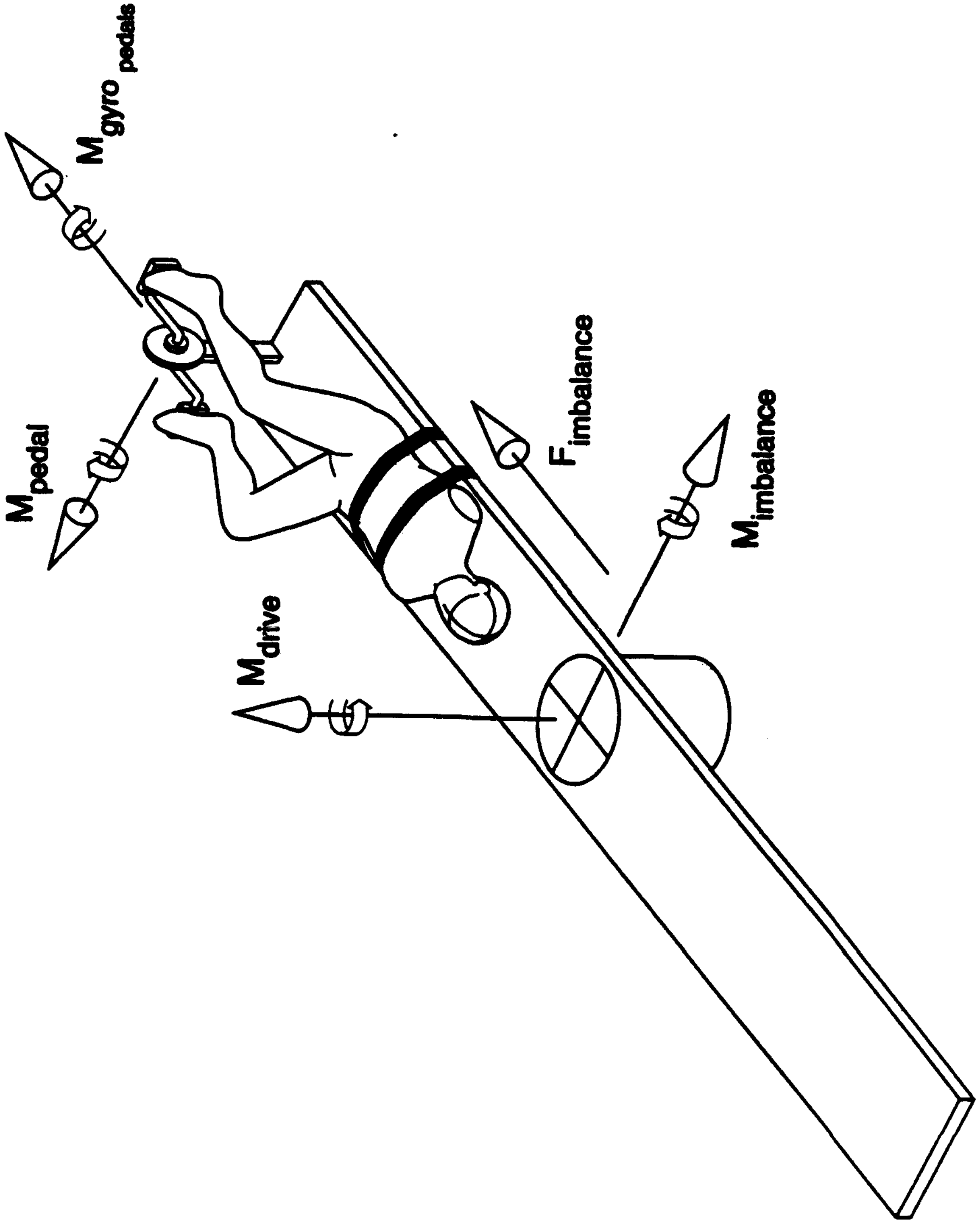


Figure 4-29. Moments and Forces on the Centrifuge



Drive and gyroscopic moment balancing could be provided by a constant speed ratio, variable inertia mechanism comprised of two masses propelled along rails. With a speed ratio of 10, the total mass of the counterinertia would only be approximately 12 kg, excluding the mass and inertia of the support structure. Another concept that would permit varying the mass of the counterinertia mechanism could feature a circular tube with spokes that would be filled with fluid as necessary to achieve the proper inertia. One difficulty with implementing a fluid-filled tube is withdrawing fluid from the tube against the radial acceleration of the device.

SECTION 5.0

5.0 Physiologic Effects of Centrifugation.

The radial acceleration generated by the centrifuge is capable of stimulating the cardiovascular system of riders, however the high angular rates may create an environment with high potential for disorientation.

5.1 Crewmember Motion Sickness and Disorientation.

On earth, a combination of otolith, visual, and somatosensory inputs enable a person to form a perception of linear acceleration and orientation with respect to the acceleration. Similarly, visual and somatosensory cues, as well as input from the semicircular canals, are processed to form an estimate of angular motion and orientation.

The human vestibular system is located bilaterally in the inner ears. It is comprised of the semicircular canals and saccular and utricular otolith organs. The function of the otoliths and canals is to sense motion of the head and relay this information to the brain. Without thoroughly describing the anatomy and operation of these sets of organs, it is well known that the otoliths are stimulated by the so-called specific force, defined as the linear acceleration of the head summed with the effect of gravity, and orientation of the head with respect to the specific force. The semicircular canals respond primarily to the resultant angular acceleration stimuli.

In the microgravity of earth orbit, a crewmember lacks the visual and gravitational cues that are usually used to define his perception of his orientation and motion. The lack of these stimuli and cues may contribute to a crewmember's susceptibility to space sickness, or Space Adaptation Syndrome. In space, contrary to on earth, the static orientation of an astronaut's head does not influence the activity of the otoliths. The nonexistence of the customary 1 G bias creates linear acceleration stimuli that can change direction rapidly even without significant head movement.[19] It is, in fact, the existence of abnormal otolith signals in the presence of seemingly normal canal and visual information that is generally regarded as a source of space motion sickness.[20]

Astronauts aboard the Space Shuttle/Spacelab 1 mission, in 1983, recorded their observations and qualitative measures of space sickness throughout the flight.[21] Most of the astronauts felt some discomfort from any type of motion, although attempting to limit head motions helped alleviate the condition.[21] Other studies hypothesize that astronauts exposed to microgravity experience a change in interpretation of sensory afferent signals, particularly from the otoliths.[19,22] Subjects participating in studies involving angular acceleration displayed no significant change in the threshold of detection.[22]

Since the short radius centrifuge conceived for space application produces rapid angular motion, it is conceivable that centrifugation could cause disorientation or space sickness. In an interesting paradox, the centrifuge might reduce the potential for space sickness by creating a radial acceleration to stimulate the otoliths, while it increases the potential for cross-coupled angular motions which cause disorientation and discomfort. In this section, we examine the potential for disorientation during short radius centrifugation and develop mathematical models for the response of the vestibular system to acceleration stimuli. A general treatment of astronaut disorientation in microgravity has not been attempted.

5.1.1 Cross-Coupling Illusion.

A disadvantage of short radius centrifugation is the increased susceptibility to the cross-coupling illusion due to the high angular rate of the centrifuge. This illusion can arise from simultaneous stimulation of the vestibular apparatus by angular motion in two planes. The vestibular system normally senses angular motion in multiple planes accurately and with no discomfort or disorientation. However, because of the dynamics of the semicircular canals, discussed in Section 5.1.2, there is potential for disorientation when the vestibular system is exposed to a long duration constant magnitude angular velocity. In this case, movement of the head relative to the existing plane of angular velocity may produce the perception of rotation about a different axis. The illusion of angular motion produced by these events is termed the cross-coupling illusion, and is sometimes referred to as Coriolis illusion.

This phenomena is familiar to aircraft pilots. During an extended turn in an aircraft, a pilot becomes acclimated to the motion so that any abrupt head movements may be disorienting. Similarly, motion on existing centrifuges is known to provoke this disorientation. Centrifuge subjects learn to avoid abrupt head movements or acclimate to the sensations. The lack of a visual reference can make the illusion even more prevalent in centrifugation.

The extremely short radius of a space-based centrifuge implies a large angular velocity would be required to achieve appropriate radial accelerations. Subjects that become acclimated to this motion may be prohibited from making even small motions of the head. In fact, research on the effect of acceleration in ameliorating or worsening the illusion indicates zero-G cross-coupling motion is less discomforting than the same motions at 1.8 G.[23]

The traditional Coriolis acceleration is not a stimulus for the cross-coupling illusion. The illusion is stimulated by the presence of cross-coupled angular velocities, also called the Coriolis *angular* acceleration. Consider the expression for the angular velocity of the centrifuge subject. If the subject is permitted to move his head relative to the centrifuge, the total angular velocity vector of his head is:

$$\boldsymbol{\omega}_{H} = \boldsymbol{\omega}_{SS} + \boldsymbol{\omega}_{C/SS} + \boldsymbol{\omega}_{H/C} \ . \qquad [5\text{-}1]$$

Differentiating, the angular acceleration of the subjects' head is:

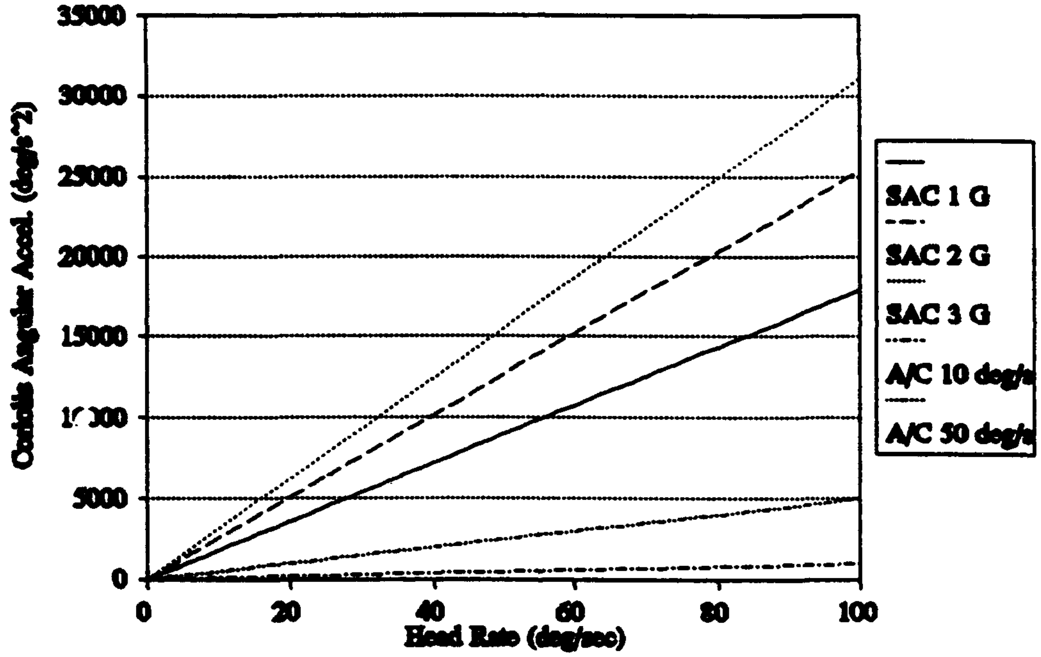
$$\boldsymbol{\alpha}_{H} = \boldsymbol{\alpha}_{SS} + \boldsymbol{\alpha}_{C/SS} + \boldsymbol{\alpha}_{H/C} + \boldsymbol{\omega}_{SS} \times \boldsymbol{\omega}_{C/SS} + \boldsymbol{\omega}_{SS} \times \boldsymbol{\omega}_{H/C} + \boldsymbol{\omega}_{C/SS} \times \boldsymbol{\omega}_{H/C} \ . \qquad [5\text{-}2]$$

The Coriolis angular acceleration represents those terms generated by the vector cross product of angular velocities. Since the angular motion of the space shuttle is likely to be a slow rate, the Coriolis angular acceleration term of most concern is:

$$\alpha_{Coriolis} = \omega_{C/SS} \times \omega_{H/C} \; . \qquad [5\text{-}3]$$

To quantitatively measure the potential for the cross-coupling illusion on the short arm centrifuge, Figure 5-1 displays a comparison of the angular velocities of a subject located 100 cm from the center of a centrifuge at different radial accelerations to a pilot of an aircraft with different turning speeds.

Figure 5-1. Comparison of Coriolis Angular Acceleration for a Centrifuge Subject and Aircraft Pilot



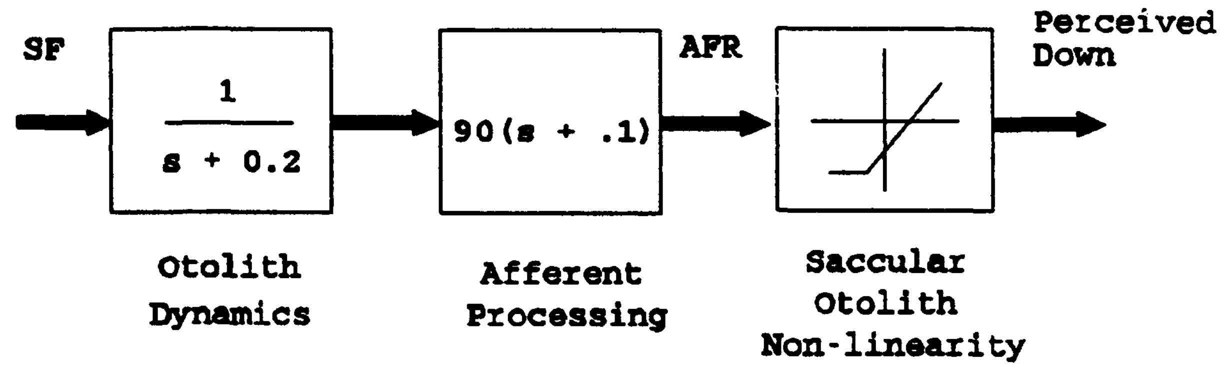
From Figure 5-1, it is obvious that the high angular velocity of the centrifuge makes the magnitude of the potential stimulus larger than during typical aircraft turns with the same head motions. Whether the increased stimulus translates into a more discomforting sense of disorientation is unknown, but would not be surprising. Maintenance of head orientation with respect to the centrifuge arm may therefore prove to be particularly important.

5.1.2 Vestibular Model.

Both sets of organs can be mathematically modelled as independent first order dynamic systems.[34]

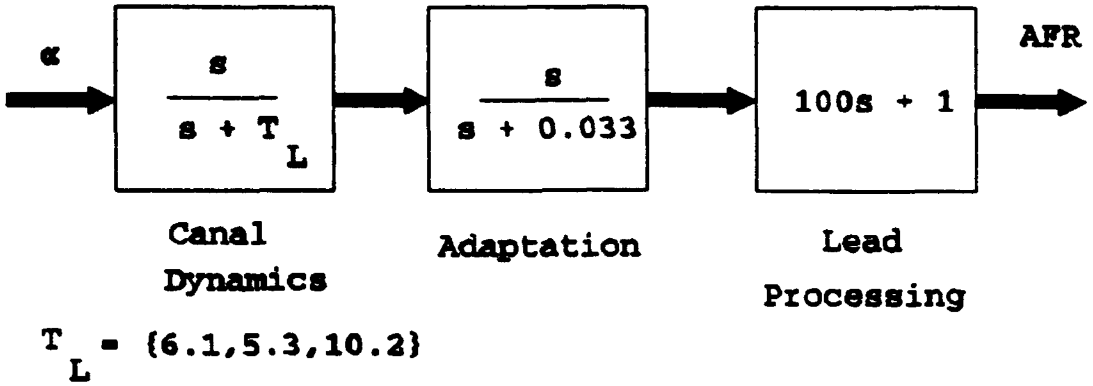
The otoliths are physically positioned in the head pitched back 25° from the head horizontal reference position. It is the projection of the gravitoinertial acceleration, called the specific force, into the pitched coordinates that stimulates the otoliths. The otoliths respond to specific force as a linear accelerometer. Accepted mathematical models of the otolith response include a first order term representing the dynamics, as well as an afferent processing term. In addition, the saccular otoliths are modelled with a non-linear term that corrects for the direction of perceived "down". The total dynamic model for the otoliths is displayed in Figure 5-2. Output of the model is a change in a bsesline rate of theoretical impulses to the brain, called the Afferent Firing Rate (AFR), in units of impulses per second. The model also outputs an estimate of the local down direction.

Figure 5-2. Otolith Dynamic Model Block Diagram



There are two sets of three orthogonal semicircular canals located on either side of the skull. As with the otoliths, the horizontal canals are pitched back approximately 25° from the horizontal. The canals act as angular acceierometers for low frequency stimuli and angular velocity transducers for high frequency stimuli. The dynamics of the canals are modelled as an overdamped torsional pendulum, with different time constants for different axis canals. An adaptation term is included by many researchers to account for the behavior in humans for the canals to become less sensitive to an angular acceleration stimulus over time. The final term is a lead processing term to account for rate sensitivity. The complete mathematical model is displayed in Figure 5-3. Again, output from the model is a theoretical afferent firing rate.

Figure 5-3. Semicircular Canal Dynamic Model Block Diagram



These mathematical models were implemented in C++ code to provide a tool for predicting the response of a crewmembers' vestibular system to acceleration stimuli.

5.1.3 Theoretical Vestibular Output Simulation.

To explore the theoretical response of the vestibular system to space centrifugation, a simulation was conducted of a typical 2 G acceleration profile. Figure 5-4 displays the tangential, radial, and total acceleration of a 95 percentile male subject with head positioned at 100 cm. The simulation consists of three segments: (1) a 30 second segment with the centrifuge at rest; (2) a 60 second segment where the crewmember is accelerated to 2 G; and, (3) a 60 second deceleration and rest segment. Figure 5-5 displays the angular velocity and angular acceleration of the subject during the simulation.

The theoretical otolith and semicircular canal responses to the centrifugation are plotted in Figures 5-6 and 5-7. During the rest segment, both the otoliths and canals transduce a zero signal. Once the centrifuge reaches a constant radial acceleration, the otoliths transduce a steady state signal. The canals become acclimated to the steady angular velocity of the second segment of the simulation, and the response decays nearly to zero. Only the x-axis canal response is plotted because the y-axis and z-axis canals are unstimulated throughout the simulation.

The subject perceived down vector is displayed in Figure 5-8. Due to the non-linear response of the saccular otolith, the perceived down direction is not along the subject z-axis. Figure 5-9 shows the angle between the perceived down and the subject z-axis. In microgravity, with no other reference cues, the subject perceives down to be angled 25° forward of his spinal axis. When the centrifuge accelerates and decelerates through 1 G, the perceived down becomes more closely aligned with the subject Z-axis. Exposed to earth's gravity in an upright position, the vestibular model would yield a perceived down vector with no x or y components and a unit z output.

Figure 5-4. Tangential, Radial, and Total Acceleration of Centrifuge Subject at 100 cm.

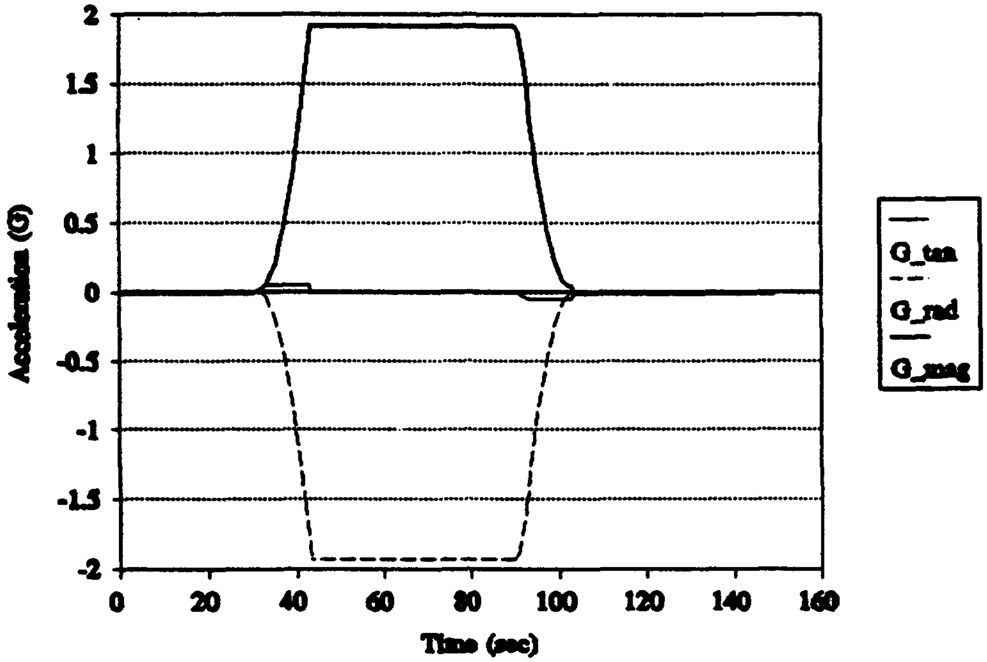


Figure 5-5. Subject Angular Velocity and Acceleration

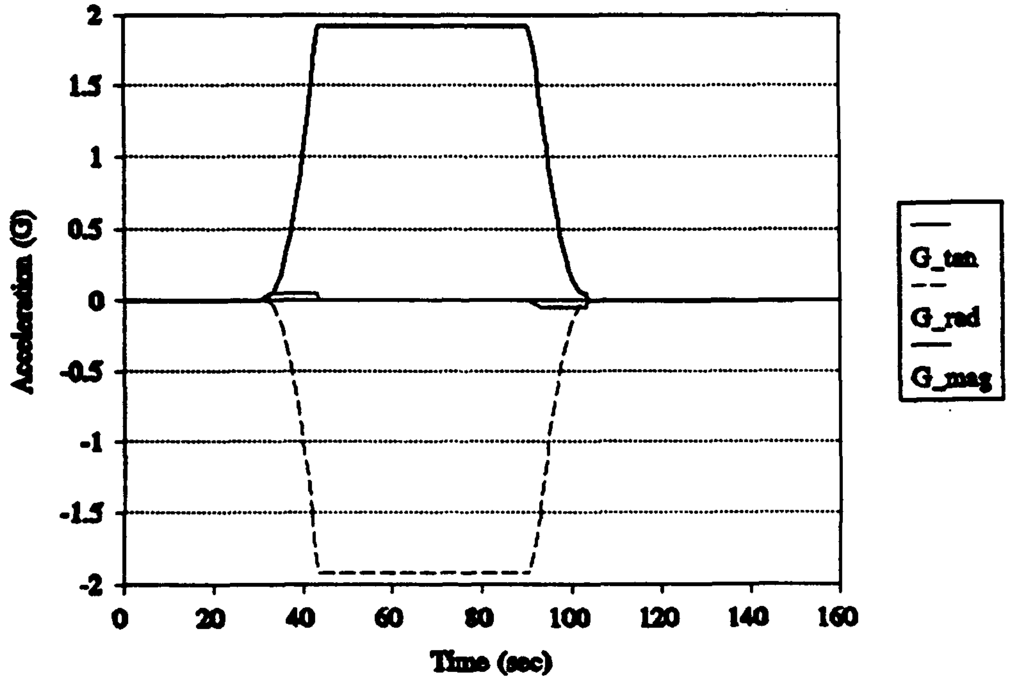


Figure 5-6. Otolith Response to 2 G Profile

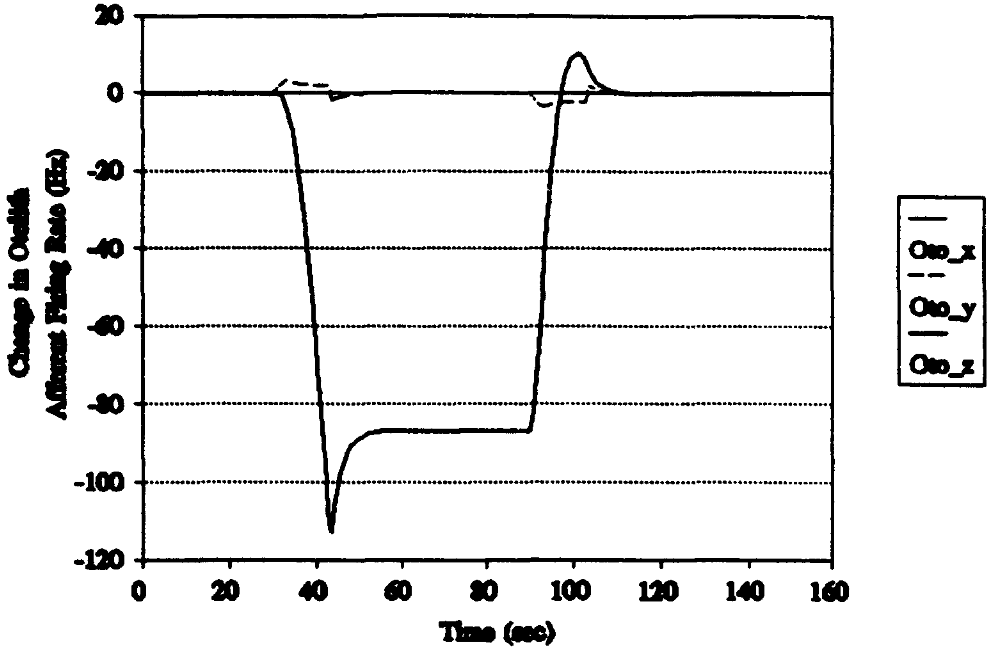


Figure 5-7. Semicircular Canal Response to 2 G Profile

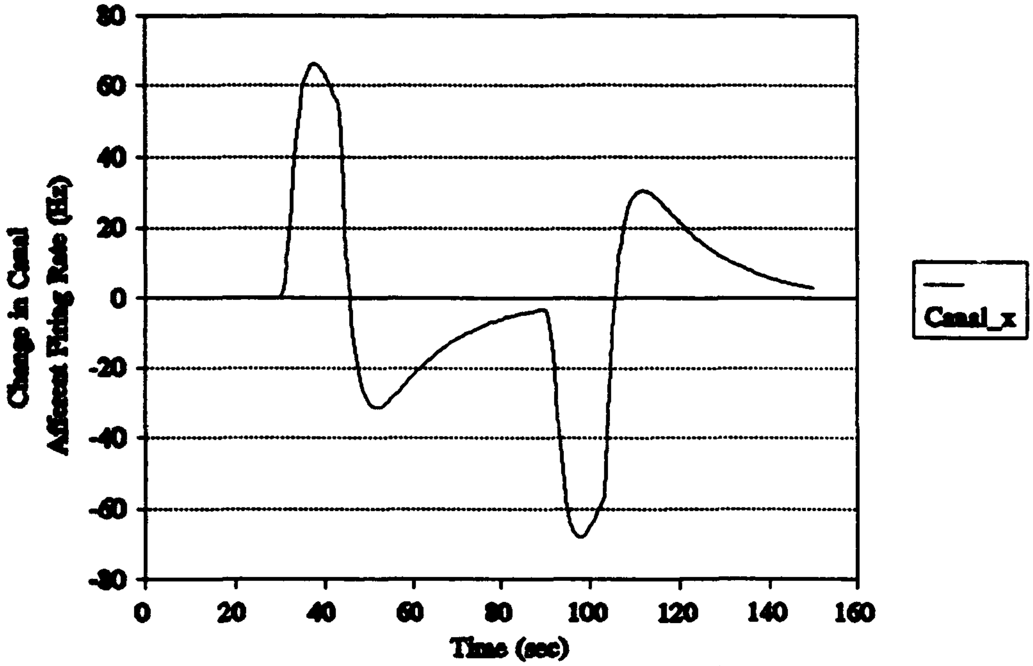


Figure 5-8. Subject Perceived Down Vector

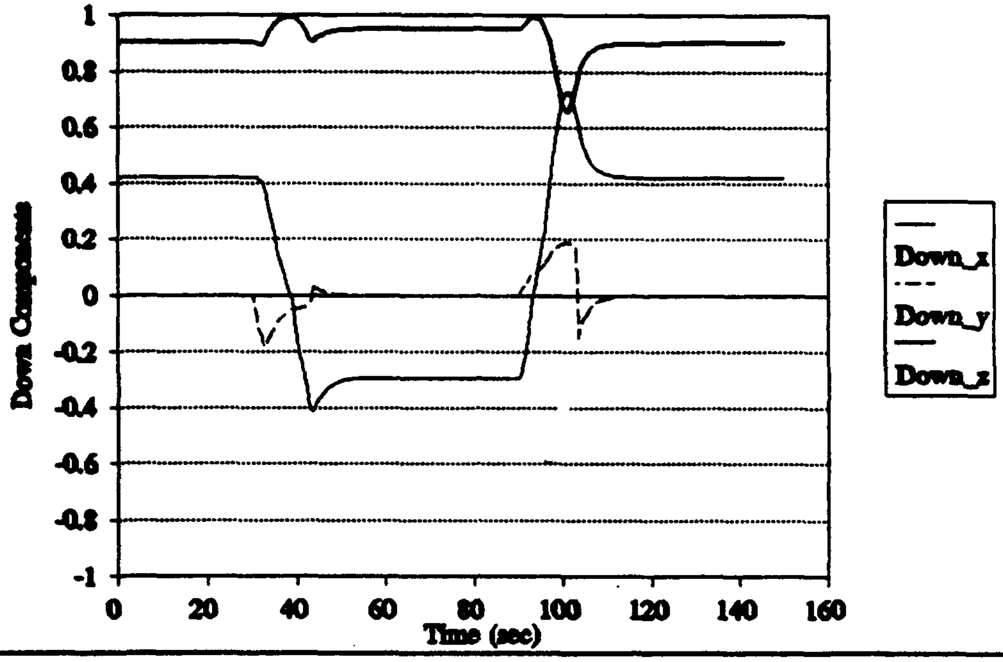
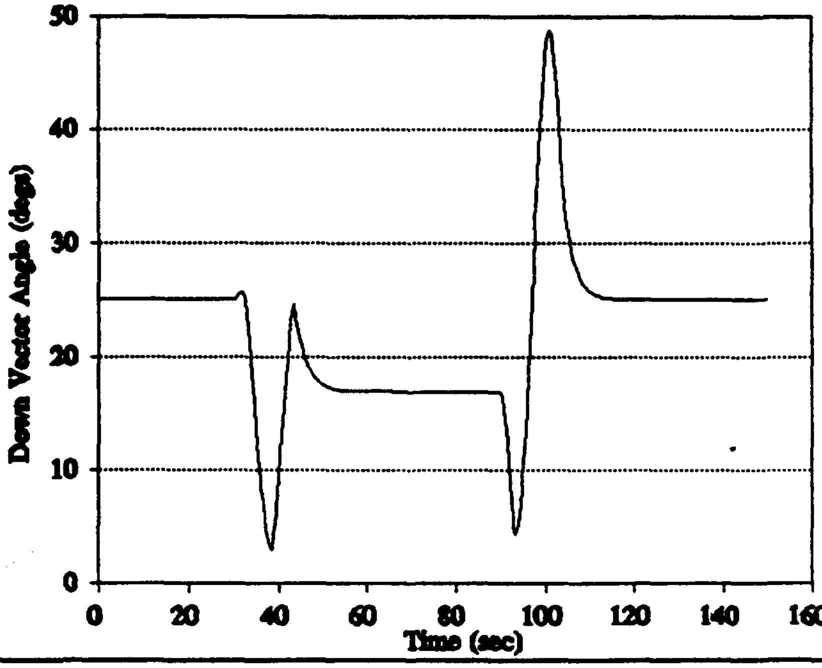


Figure 5-9. Angle Between Perceived Down and Subject Z-Axis



It is afusr the canals have adapted to the angular velocity that the cross-coupling illusion becomes a concern. From Figure 5-7, it is obvious the signal from the canals normally decays during constant angular motion. Head motion even after the decay might be disorienting.

One way to prevent disorientation might be to continually stimulate the canals to prevent the acclimation process. Frequent head motion of the subjects, either voluntarily or through a moving head cushion device, could serve to insure the canals correctly transduce the rotation.

5.2 Cardiovascular Effects.

Exposure to microgravity disrupts the equilibrium point of many physiologic processes, including some cardiovascular system functions. Although the deconditioning of the cardiovascular system is not discussed in detail in this report, short radius centrifugation has been shown to stimulate the heart.[8] In this section, we discuss the significance of the acceleration gradient formed by the short radius centrifuge, describe a mathematical model for the cardiovascular system, and report the results of simulations of the cardiovascular model.

5.2.1 Acceleration Gradient Effect.

Any rotating object generates a radial acceleration at a point that is proportional to the distance from the center of the object. A body subjected to the rotation will experience an acceleration gradient because *of the difference* in acceleration at different radii. This gradient is most important at short radii, such as on a short arm centrifuge.

Recalling Equation 4-5, the acceleration gradient with respect to centrifuge radius is a constant for a constant angular velocity:

$$\frac{dG}{dr} = \frac{\omega_C^2}{g_0} . \qquad [5\text{-}4]$$

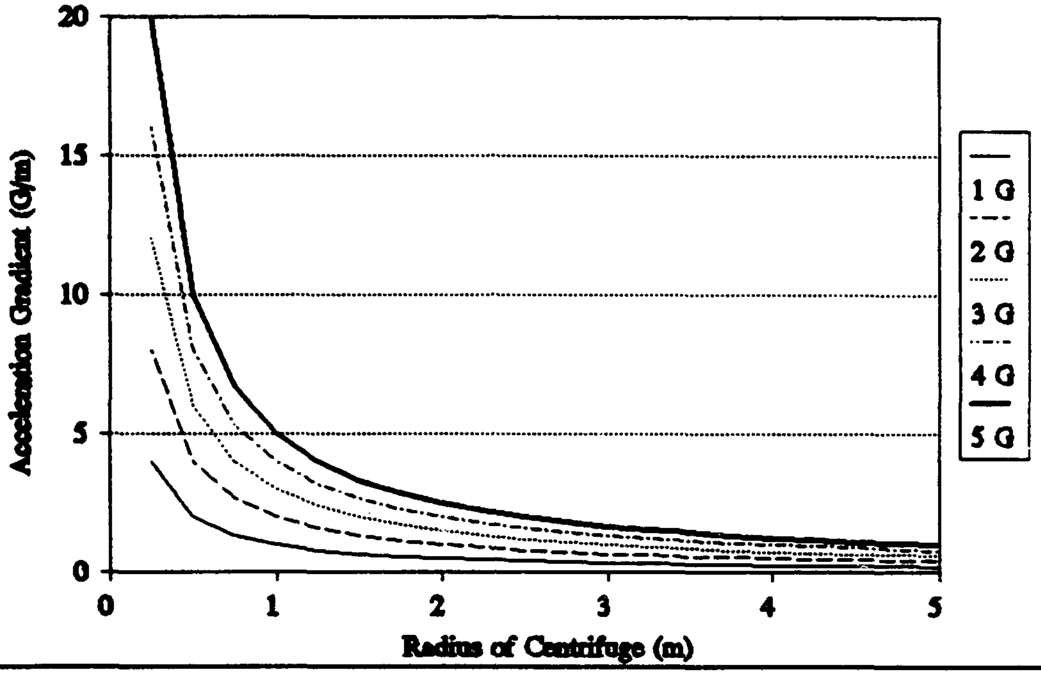
It is usually the acceleration of the crewmember, however, that is used to select an operating condition for a centrifuge. Thus, it is more interesting to compare the acceleration gradient for centrifuge riders that have the same acceleration magnitude at the top of their head. In this case, Equation 5-4 can be replaced with the desired acceleration at the head divided by the radius from the center of the centrifuge:

$$\frac{dG}{dr} = \frac{G_d}{r} , \qquad [5\text{-}5]$$

where G_d is the desired acceleration in G.

The acceleration gradient for different radii on a centrifuge are plotted in Figure 5-10. The five curves represent constant acceleration curves from 1 G to 5 G. As expected, the acceleration gradient is highest for short radii and large accelerations.

Figure 5-10. Acceleration Gradient at Different Radii



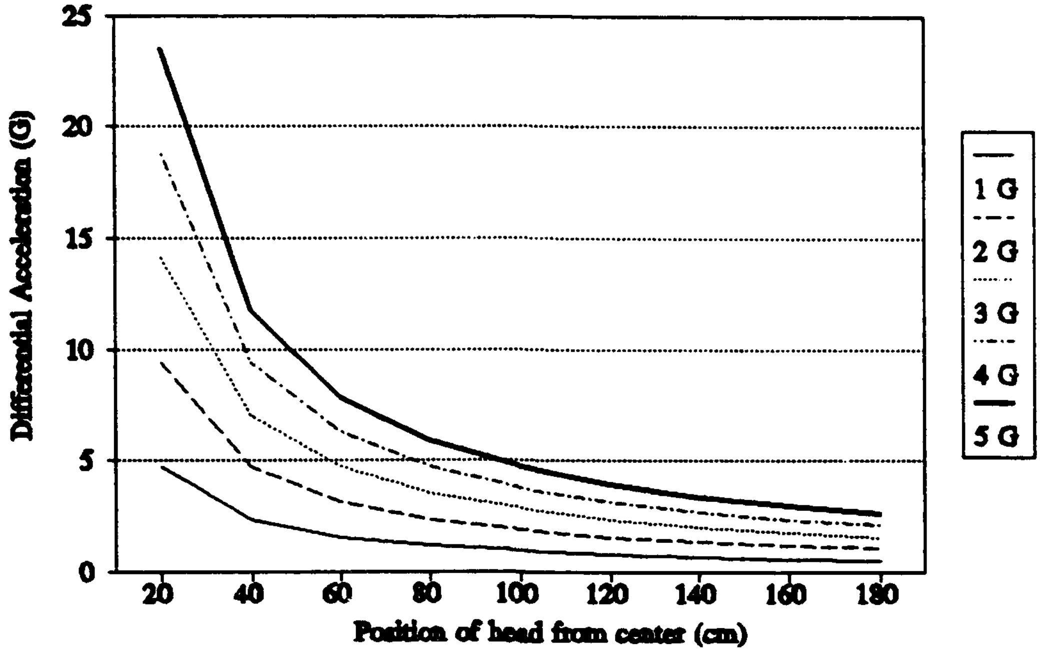
The differential acceleration on a centrifuge for a 95 percentile male rider, in units of earth's gravity, is displayed in Figure 5-11. Recall that the distance from the top of the head to the thighs-up feet position is approximately 99.5 cm. Figure 5-11 displays the difference in acceleration between the top of the head and the feet for different accelerations at the head.

The differential acceleration can also be expressed in the form of a percentage by dividing the change in acceleration across the body by the heart-level acceleration:

$$\%\Delta G = \frac{G_{head} - G_{feet}}{G_{heart}} \quad . \qquad [5\text{-}6]$$

The percentage change in acceleration depends only on the distance of the riders head from the center of the centrifuge and not the acceleration magnitude at the head. For a rider positioned 100 cm from the spin axis, the percentage change is approximately 10%.

Figure 5-11. Differential Acceleration for Different Rider Positions

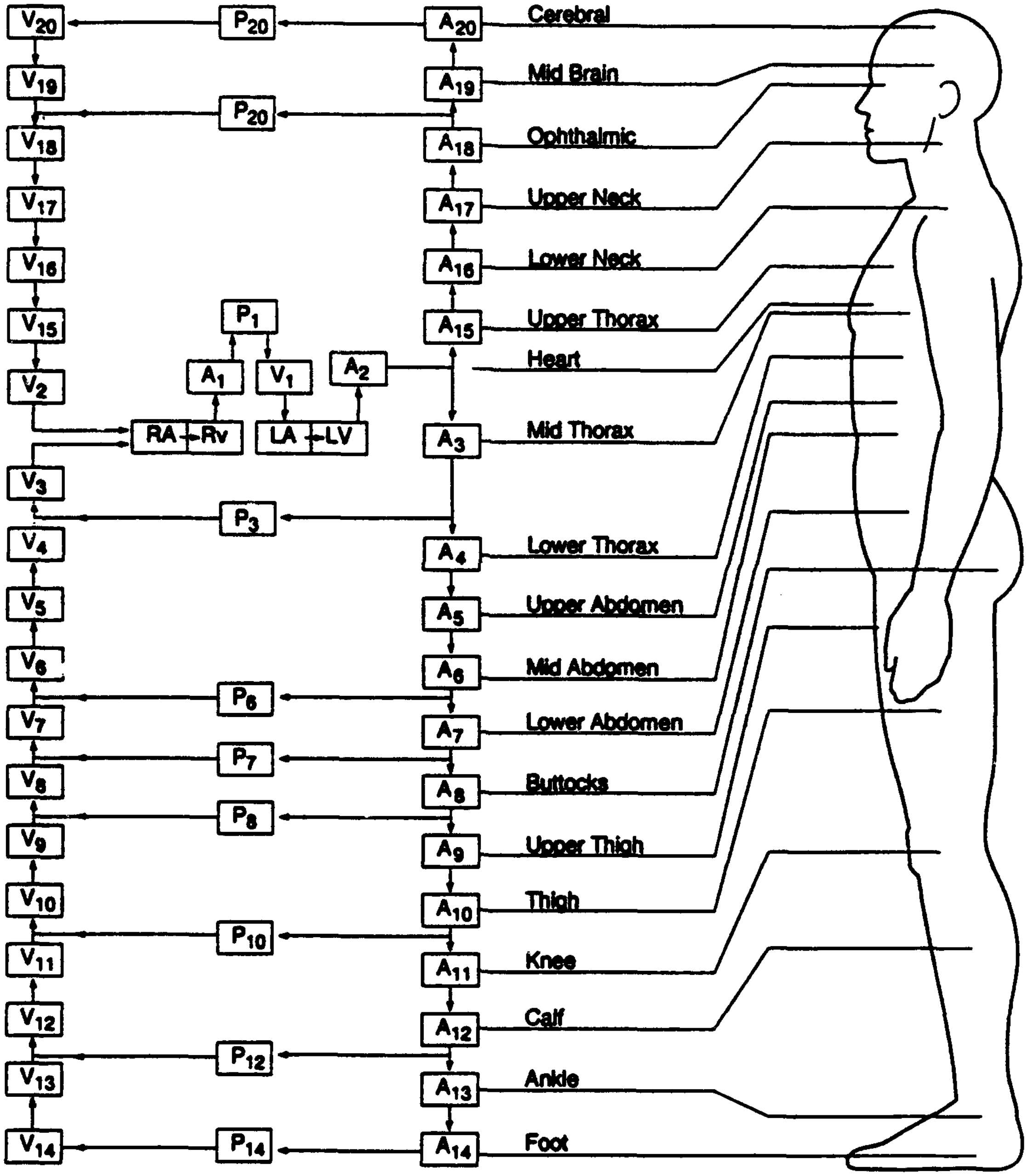


5.2.2 Cardiovascular Model and Simulation.

The cardiovascular model described herein parallels the development presented in a paper by White, et al.[25] It also draws from papers by Jaron, et al.,[26,27] whose approach was similar to that presented in earlier papers by Rideout,[28] Snyder,[29] and Avula.[30] All of these authors cited the earlier work of Wormersly.[31] Many of the parameter values for the physical properties of the vascular segments were taken from these papers as well as Fung[32] and Bergel.[33] However, some of the parameters listed in Jaron[26] were clearly in error and some of those employed by other authors were not useful because of differences in the layout of their models and ours. Where new parameters were required, they were derived to yield generally acceptable flow/pressure/volume and compliance characteristics of the various cardiovascular subdivisions. In particular, flow, pressure, and compliance data from Burton,[34] and Guyton,[35] were employed.

Figure 5-12 displays the general schematic for the model compartments and their connections. The model describes the spatial and temporal variation in the mean blood pressure along the z-axis of the body. The present implementation neglects the non-linear and convective terms in the Navier-Stokes Equation as well as the complication of radial flow (flow perpendicular to the vessel wall). The flow is assumed to be laminar except in the ascending an descending aorta where fluid flow resistances were multiplied by 33 to account for increased pressure losses in turbulent flow.[28]

Figure 5-12. Cardiovascular System Model Compartments and Connections



The following set of coupled differential equations were solved for each arterial and venous vascular segment.

$$\frac{dP(t)}{dt} = \frac{1}{C}\cdot(Q(t)_{in} - Q(t)_{out}) + R_w\cdot(Q(t)_{in} - Q(t)_{out}) \ , \tag{5-7}$$

$$\frac{dQ(t)}{dt} = \frac{1}{L}\cdot(P(t)_{in} - P(t)_{out} + P_{G_z} - R\cdot Q(t)) \ , \tag{5-8}$$

$$\frac{dr(t)}{dt} = \frac{1}{2\cdot\pi\cdot r(t)\cdot l}\cdot(Q(t)_{in} - Q(t)_{out}) \ . \tag{5-9}$$

Where,

t	=	Time	[sec].
r	=	Radius of vascular segment	[m].
l	=	Length of vascular segment	[m].
P	=	The pressure in the segment	[Pa].
Q	=	The segmental volume flow	[$m^3\cdot sec^{-1}$].
C	=	The capacitance of the segment	[$m^3\cdot Pa^{-1}$].
L	=	The inertance of the segment	[$kg\cdot m^{-4}$] or [$Pa\cdot sec^2\cdot m^{-3}$]
R	=	The viscous flow resistance in the segment	[$Pa\cdot sec\cdot m^{-3}$].
R_w	=	The vessel expansion resistance	[$Pa\cdot sec\cdot m^{-3}$].
P_{Gz}	=	The hydrostatic pressure difference across the segment because of gravity	[Pa].

And, the following approximations for R, L, and C were taken from Rideout, et al.[28]

$$R = \frac{81\cdot\mu_0\cdot l}{8\cdot\pi\cdot r^4} \ , \tag{5-10}$$

$$L = \frac{9\cdot\rho_0\cdot l^2}{4\cdot V} \ , \tag{5-11}$$

$$C = \frac{3\cdot r\cdot V\cdot l}{2\cdot E\cdot h} \ , \tag{5-12}$$

$$R_w = \frac{0.002}{C} \ . \tag{5-13}$$

Where,

E = Young's modulus for vessel wall [Pa].
h = Vessel Wall thickness [m].
ρ_0 = density of blood [$kg \cdot m^{-3}$].
μ_0 = viscosity of blood [$N \cdot sec \cdot m^{-2}$].

Finally,

$$P_{G_z} = \rho_0 \cdot G_z \cdot g_0 \cdot l \cdot Cos(\theta) \ . \qquad [5\text{-}14]$$

Where,

G_z = The z-axis "G-level" in units of earth's gravity [unitless]
g_z = The earth's gravitational acceleration [$m \cdot sec^{-2}$]
θ = The angle between the segment and the z-axis [radians].

For each vascular segment, Equations 5-7, 5-8, and 5-9 must be solved simultaneously. There are 20 arterial and 20 venous vascular segments (the pulmonary circuit is segment 1). There are 10 peripheral capillary bed segments modelled by differential equations for pressure and flow. The peripheral capillary resistance and capacitance are modelled as "T" circuits with the peripheral capacitance in parallel with an inlet and outlet resistance which represent the resistance of the arteriole and venule respectively. The pulmonary circulation is modelled as a single peripheral bed fed by a single arterial segment and drained by a single venous segment. The chambers of the heart are modelled as variable capacitances separated by one-way valves. The pulmonic and aortic valves are also modelled as one-way valves. The pressure reference for the model is located at the tricuspid valve which is presumed to track intrathoracic pressure. The pressures and flows in the various segments are coupled by their spatial connections. The z-axis coordinates for locating the origin and termination of each segment were based on a 177 cm tall standing man. Including the equations for pressure and flow in the heart, there are 174 differential equations in the model. The solution was computed using a Runge-Kutta numerical integration algorithm.[36]

The initial conditions for segmental flow were set based on a steady-state solution for the model at 1 G_x (transverse) corresponding to a supine posture. Initial pressures were set by computing the hydrostatic pressure at each segment corresponding to the initial local *z*-axis acceleration. Posture is adjusted by setting the appropriate initial orientation angle for each segment relative to the *z*-axis acceleration. Postural and acceleration changes during a simulation may be made by adjusting the orientation angles and/or acceleration during the simulation. This can be most easily accomplished by adding time varying profiles for G_z and the θ's from which the P_{Gz} values for the flow derivatives are calculated. However, the current version of the model does not include provisions for simulating the effects of the short term cardiovascular reflexes responsible for controlling the blood pressure and distribution of blood flow. Therefore, no attempt was made to match transient responses of the model to those expected in the response of the true cardiovascular system. Rather, the model was coded so that the cardiac output was maintained at a constant value (5.4 liters·min^{-1}) until the regional blood flow values returned to their initial values. The resulting steady-state pressures or "equilibrium pressures" in the various segments represent the pressures required to maintain "normal" blood flow through the vascular and peripheral segments. The results of the two simulations described below illustrate the results of this process.

Presumably, the short arm centrifuge should produce cardiovascular pressure similar to those experienced by man at +1 G_z in the upright posture. To compare the blood pressure profile produced by the space based short arm centrifuge to that seen in a normal upright posture on earth's surface, two simulations were computed: one in which a constant +1 G_z was applied to all segments, and a second simulating the acceleration produced by the short arm centrifuge in microgravity. The profile employed for the space based centrifuge produced approximately +1 G_z at the top of the head and approximately +3 G_z at the feet. The orientation angles for the hips and thigh were adjusted to match the supine pedalling posture. Figure 5-13 and Figure 5-14 compare the equilibrium arterial and venous pressures estimated by the model for the two situations. As shown, at +1 G_z both the arterial and venous pressures increase smoothly below the heart (the reference pressure) and decrease smoothly above the heart. In contrast, the pressures in the theoretical space based centrifuge rider show a tendency to plateau in the hip and thigh area (segments 9 and 10) because the acceleration is transverse to those segments. The only changes in the pressure are due to flow resistances which are small in comparison to the gradients produced by acceleration. In addition, the pressures are higher below the heart and lower above the heart as would be expected because the acceleration field is changing in the *z*-axis direction.

The results of this simulation show only very small differences in pressure between the between the cardiovascular pressures produced in the space based centrifuge and those seen in the upright man at +1 G_z. Moreover, the effect of exercise by the centrifuge rider would undoubtedly reduce the venous pressures via the "venous pump." There is no provision for such a muscular pumping action in the current version of the model, thus this effect is not apparent in the simulation comparisons in Figure 5-13 and Figure 5-14.

Two additional simulations were computed to compare the operation of the short arm centrifuge on earth to its operation in microgravity. The same *z*-axis acceleration field as described above, was employed in two simulations: one with 0 G_x to simulate space base operation of the short arm centrifuge, and one with +1 G_x to simulate its operation on earth. Figure 5-15 and Figure 5-16 compare the steady state arterial and venous pressures estimated for the two situations. As shown, the earth based operation and the space based operation produce virtually identical pressure profiles above the hips. However, below the hips, the earth based profile results in higher arterial and venous pressures because the pressures at the hip segment must be higher to move blood against the +1 G_x acceleration which is oriented along the hip and thigh segments. The venous pressures are correspondingly higher in the earth based centrifuge although the pressure gradient in the venous segments below the hip is much higher than in the space based centrifuge. The comments regarding the effects of exercise on the venous pressures also apply to the comparison between the space based and earth based profiles. It is likely that the venous pumping action of exercise would reduce venous pressures in both situations and the differences between the two.

In summary, the cardiovascular simulations have served to add support to the efficacy of the use of the short arm centrifuge to approximate the steady-state cardiovascular pressure profiles seen in an upright human on earth. It appears feasible and desirable to extend the Phase I model in Phase II to include the short term physiologic control and the effects of exercise on cardiovascular flows and pressures. This will enable the model to more faithfully simulate the true response of the cardiovascular system. An enhanced Phase II cardiovascular model could be validated against earth based experimental data and then employed to predict the differences between profiles conducted on earth and in microgravity.

Figure 5-13. Equilibrium Arterial Pressures
Upright 1 Gz vs Space Based

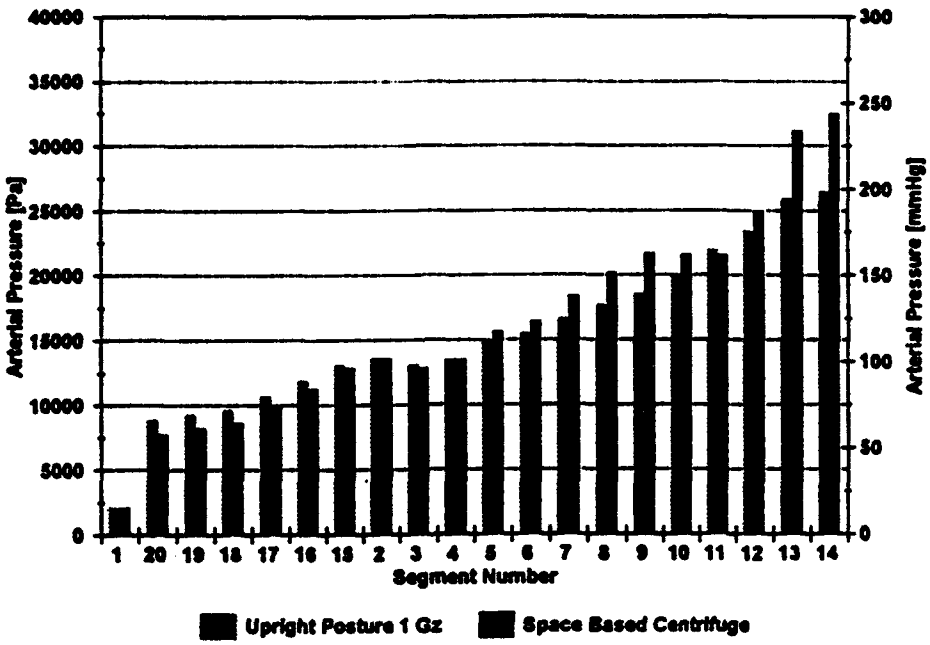


Figure 5-14. Equilibrium Venous Pressures
Upright 1 Gz vs Space Based

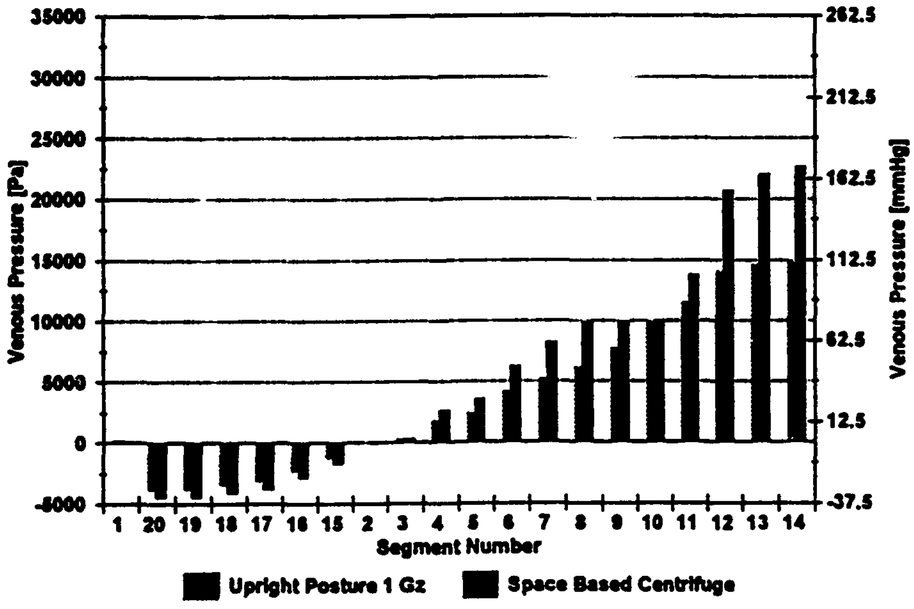


Figure 5-15. Equilibrium Arterial Pressures
Short Arm Centrifuge

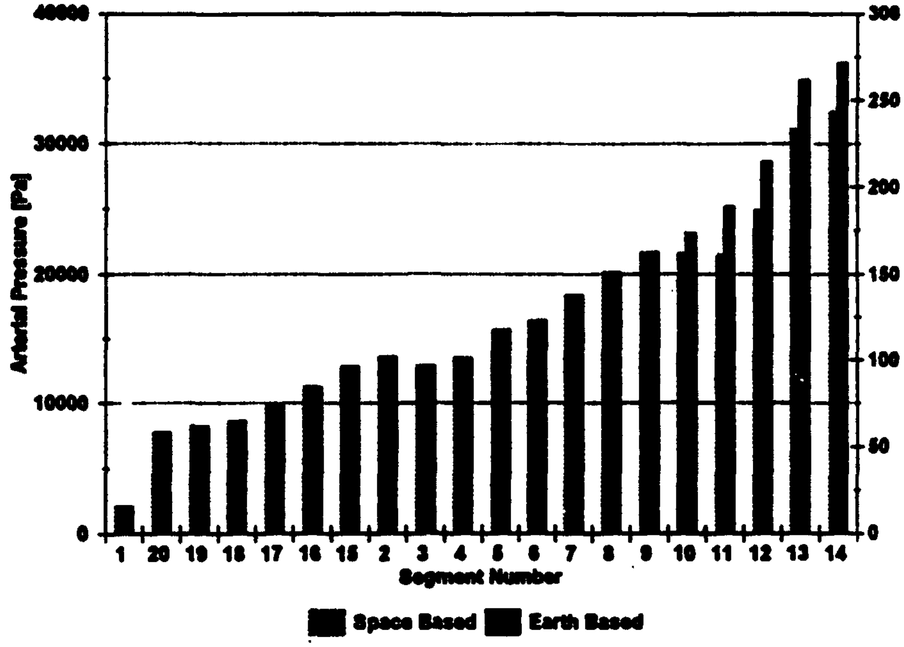
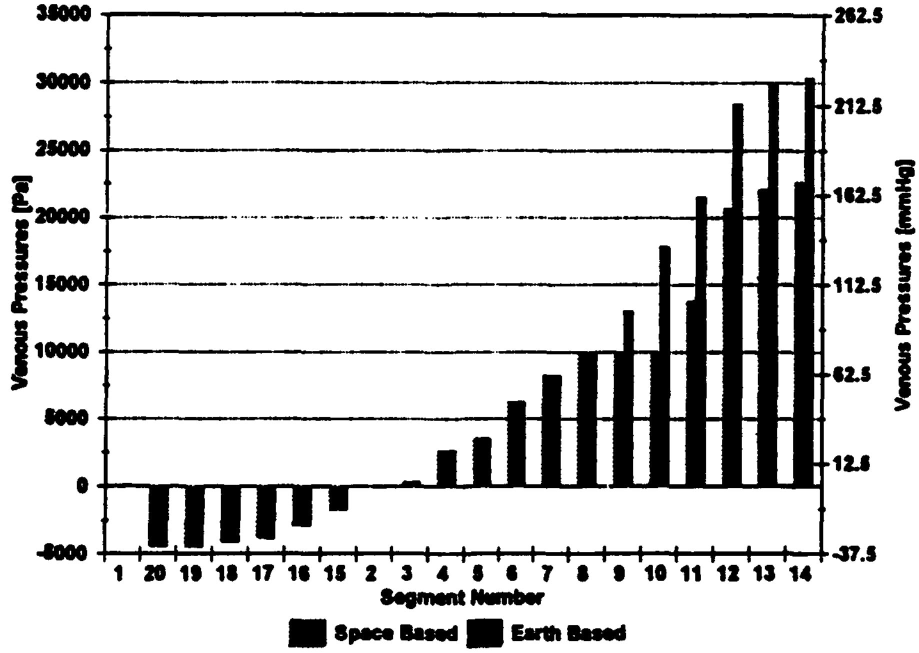


Figure 5-16. Equilibrium Venous Pressures
Short Arm Centrifuge



REFERENCES

REFERENCES

1. Meeker, LJ; Isdahl, W. Parametric Design Study for a Small Radius Centrifuge for Space Application. Presented at the 10th IAA Man in Space Symposium, April 1993, Tokyo, Japan.

2. Burton, RR; Smith, AH. Adaptation to Acceleration Environments. IN PRESS: Handbook of Physiology - Adaptation to the Environment. Chapter 36. American Physiology Society.

3. Burton, RR. A human-use centrifuge for space stations: proposed ground-based studies. Aviat. Space Environ. Med. 579-582; 1988 June.

4. Harding, R. Survival in Space. Routledge, London: 1989.

5. White, WJ; Nyberg, JW; White, PD; Grimes, RH; Finney, LM. Biomedical Potential of a Centrifuge in an Orbiting Laboratory. SM-48502; SSD-TDR-64-209-Suppl: 129 pp. 1965 July.

6. Diamandis, PH. Use of a 2-Meter Radius Centrifuge on Space Station for Human Physiologic Conditioning and Testing. Proceedings of the 8th Princeton/AIAA/SSI Conference, May 6-9, 1987, Princeton, NJ. "Space Manufacturing 6 - Nonterrestrial Resources, Biosciences, and Space Engineering". Washington, D.C.: American Institute of Aeronautics and Astronautics; 1987; 133-136.

7. Burton, RR. Periodic acceleration stimulation in space. Presented at the 19th Intersociety Conference on Environmental Systems, July 24-26, 1989, San Diego, CA. Warrendale, PA: Society of Automotive Engineers; 1989: 1-4. ISSN: SAE Paper #891434.

8. Burton, RR; Meeker, LJ; Raddin, JH, Jr. Centrifuges for studying the effects of sustained acceleration on human physiology. IEEE Eng Med Biol. 10: 56-65; 1991 March.

9. Burton, RR; Meeker, LJ. Physiologic validation of a short-arm centrifuge for space application. Aviat. Space Environ. Med. 63: 476-481; 1992 June.

10. Halstead, TW; Brown, AH; Fuller, CA; Oyama, J. Artificial gravity studies and design considerations for space station centrifuges. Proceedings of the 14th Intersociety Conference on Environmental Systems, July 16-19, 1984, San Diego, CA. Warrendale, PA: Society of Automotive Engineers; 1984; 1-11. ISSN: SAE Paper #840949.

11. Man-Systems Integration Standards. NASA-STD-3000. Rev. A, v. 1; 1989 October.

12. Antonutto, G; Capelli, C; di Prampero, PE. Pedalling in space as a countermeasure to microgravity deconditioning. Microgravity. 1(2): 93-101; 1991.

13. Kautz, RW; Feltner, ME; Coyle, EF; Baylor, AM. Pedaling Techniques of Elite Endurance Cyclists: Changes with increasing workload at constant cadence. IN: International Journal of Sport Biomechanics. 7(1):29-53; 1991 February.

14. Åstrand, PO; Rodalh, K. Textbook of Work Physiology - Physiological Bases of Exercise, 3rd ed. New York: McGraw-Hill; 1986.

15. Barret, C. Spacecraft Flight Control System Design Selection Process for a Geostationary Communication Satellite. NASA Technical Paper NASA-TP-3289; 1992: 20 pp.

16. Arno, RD; Horkachuck, MJ. Research centrifuge accommodations on space station freedom. Presented at the 20th Intersociety Conference on Environmental Systems, July 9-12, 1990, Williamsburg, VA. Warrendale, PA: Society of Automotive Engineers; 1990: 1-9. ISSN: SAE Paper #901304.

17. Johnson, CC. The Biological Flight Research Facility. Proceedings of the 42nd International Astronautical Congress, October 5-11,1991, Montreal, Quebec, Canada. 1991 Oct: 8 pp. ISSN: IAF Paper 91-578.

18. Searby, N. Effect of science laboratory centrifuge of space station environment. IN: NASA, Marshall Space Flight Center, Measurement and Characterization of the Acceleration Environment on Board the Space Station. 1990 Aug; 17 pp.

19. Young, LR; Oman, CM; Watt, DGD; Money, KE; Lichtenberg, BK; Kenyon, RV; Arrott, AP. M.I.T./Canadian vestibular experiments in Spacelab-1 mission: 1. Sensory adaptation to weightlessness and readaptation to one-g: an overview. Exp Brain Res. 64: 291-298; 1986.

20. Benson, AJ; Kass, JR; Vogel, H. European vestibular experiments on the Spacelab-1 mission: 4. Thresholds of perception of whole-body linear oscillation. Exp Brain Res. 64: 264-271; 1986.

21. Oman, CM; Lichtenberg, BK; Money, KE. Space motion sickness monitoring experiment: Spacelab 1. IN: AGARD Motion Sickness: Mechanisms, Prediction, Prevention and Treatment. 1984 November; 21 pp.

22. Benson, AJ. Effect of spaceflight on thresholds of perception of angular and linear motion. Arch Otorhinolaryngol. 244: 147-154; 1987.

23. DiZio, P; Lackner, JR; Evanoff, JN. The Influence of Gravitoinertial Force Level on Oculomotor and Perceptual Responses to Coriolis, Cross-Coupling Stimulation. Aviat. Space Environ. Med. 58(9 Suppl): A218-23; 1987.

24. Bomar, JB; Pancratz, DJ; Raddin, JH; Harmony, DW; Hessheimer, M; Jacob, JB. Engineering Design Analysis of a Large Radius Track Centrifuge. Final Report for Research Conducted under U.S.A.F. Small Business Innovation Research Contract

#F41624-91-C-2002. Submitted for Publication as U.S.A.F. Armstrong Laboratory Technical Report, February, 1993.

25. White, RJ; Cronton, DG; Fitzgerald, DG. Cardiovascular Modelling: Simulating the Human Cardiovascular Response to Exercise, Lower Body Negative Pressure, Zero Gravity and Clinical Conditions. Adv. Cardiovasc. Phys. Part I: 195-229; 1983.

26. Jaron, D; Moore, TW; Bai, J. Cardiovascular Response to Acceleration Stress: A Computer Simulation. Proceedings of the IEEE, 76(6): 700-707; 1988.

27. Jaron, D; Moore, TW; and Chu CL. A Cardiovascular Model for Studying Impairment of Cerebral Function During $+G_z$ Stress. Aviat. Space Environ. Med. 55(1): 24-31; 1984.

28. Rideout, VC; Dick, DE. Difference-Differential Equations for Fluid Flow in Distensible Tubes. IEEE Trans. Bio-Med. Engr. 14(4): 171-177; 1967.

29. Snyder, MF; Rideout, VC. Computer Simulation of the Venous Circulation. IEEE Trans. Bio-Med. Engr. 16(4): 325-334; 1969.

30. Avula, XJR; Ostreicher, HL. Mathematical Model of the Cardiovascular System Under Acceleration Stress. Aviat. Space Environ. Med. 49(1):279-286; 1978.

31. Womersley, JR. An elastic tube theory of pulse transmission and oscillatory flow in mammalian arteries. Wright Air Development Report WADC-TR-56-614. 1957.

32. Fung, YC. Biomechanics. New York: Springer-Verlag; 1993.

33. Bergel, DH. The Dynamic Elastic Properties of the Arterial Wall. J. Physiol. 156: 458-469; 1961.

34. Burton, AC. Physiology and Biophysics of the Circulation. Chicago, IL: Year Book Medical Publishers, Inc.; 1965.

35. Guyton, AC. Textbook of Medical Physiology. 7th Ed. Philadelphia, PA: W.B. Saunders; 1985.

36. Schiesser, WE. The Numerical Method of Lines. San Diego: Academic Press Inc.; 1991.

APPENDIX A

Appendix A1
Inertia Calculations for Pedals and Feet

The gyroscopic moment produced by the cross-coupled motion of the pedal mechanism and the centrifuge is proportional to the inertia of the mass rotating with the pedals. Consider the subject and pedals of Figure A-1. The total effective moment of inertia of this system includes the inertia of the pedal mechanism as well as the inertia of the subject's feet and perhaps part of the lower leg. For purposes of our analysis, we will consider only approximate masses for the pedals and feet, so that the moment of inertia about the pedal axis of rotation is:

$$I_{tot} = 2 \cdot I_{ped} + 2 \cdot I_{ft} \ . \qquad \text{[A1-1]}$$

To compute the moment of inertia of the pedals, assume each pedal and support has a mass of 0.5 kg, the effective radius of rotation is 15 cm, and the centroidal inertia of the pedal is negligible. Then, the inertia is:

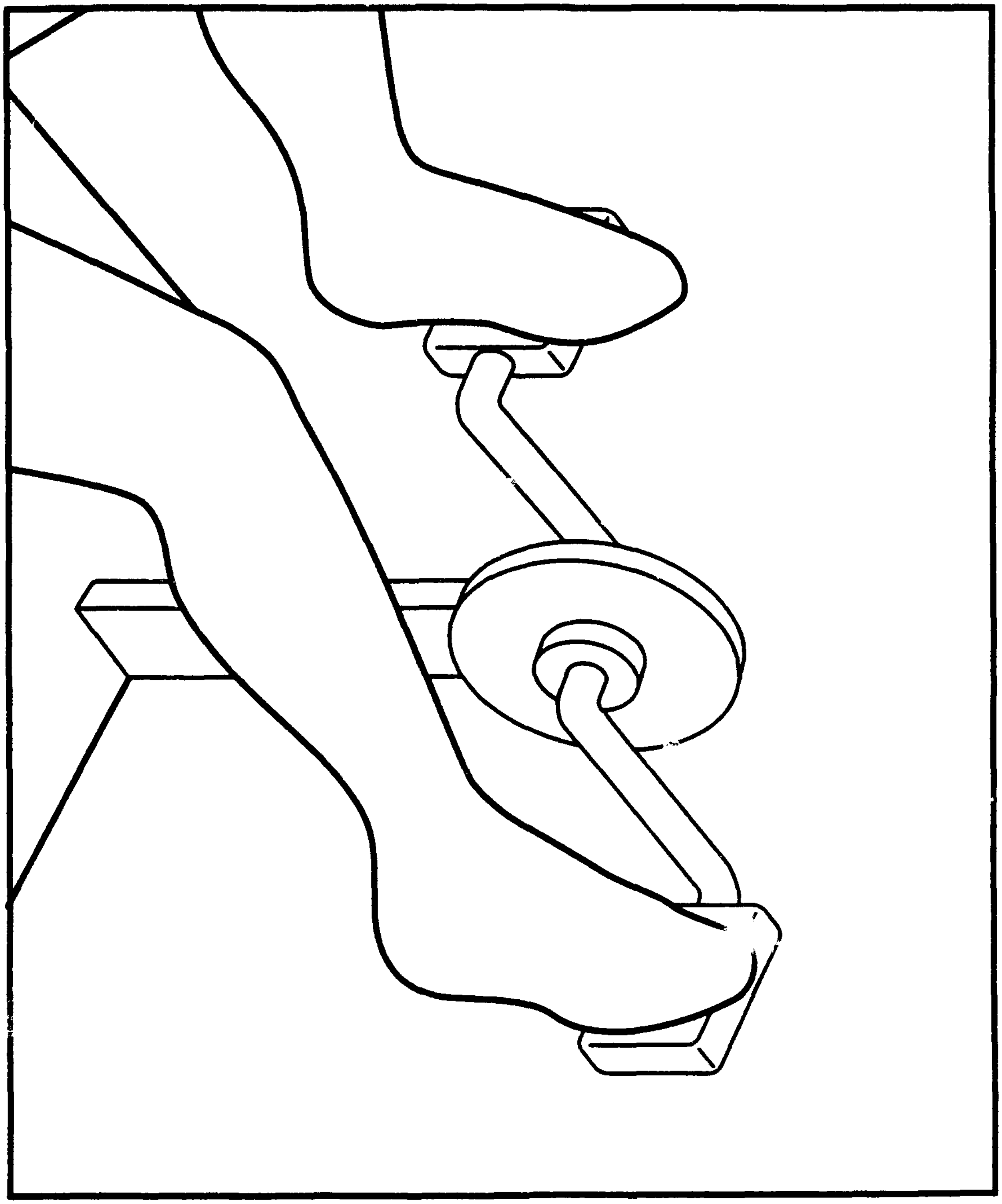
$$I_{ped} = m_{ped} r_{ped}^2 = 0.0113 kg \cdot m^2 \ . \qquad \text{[A1-2]}$$

The mass and centroidal inertia of the foot is taken from an anthropometry source.[1] It is assumed that the center of mass of each foot is located at approximately the distance of the pedal, or 15 cm. The inertia of a foot, then, is:

$$I_{ft} = \bar{I}_{ft} + m_{ft} r_{ft}^2 = 0.0269 kg \cdot m^2 \ . \qquad \text{[A1-3]}$$

Finally, substituting values from equations A1-2 and A1-3 into A1-1, the approximate moment of inertia of the rotating system is 0.0764 kg·m^2. This value could change significantly if the mass of the pedal mechanism was changed, if the crewmembers wore boots, or if the pedal arms were different lengths.

Figure A-1. Rider Feet and Pedal Mechanism Inertia



REFERENCES

1. USAARL 88-5. Anthropometry and Mass Distribution for Human Analogues. Vol. 1:Military Male Aviators, March, 1988.

APPENDIX B

APPENDIX B1
Theoretical Vestibular Response Program

B1.0 **Program Name.** VESTIB.EXE, for MS-DOS.

B1.1 **Purpose.** VESTIB computes the theoretical vestibular response of a human to angular acceleration and linear acceleration stimuli. The vestibular response consists of the otolith theoretical afferent firing rates (AFR), semicircular canal afferent firing rates (AFR), and theoretical perceived down vector. The dynamic models for the otoliths and canals are described in Chapter 5.

B1.2 **General Description of Program Flow.** On execution, VESTIB opens and reads a data file containing the x, y, and z components, respectively, of the subject angular velocity, angular acceleration, and specific force. The program then computes the incremental vestibular response to the stimuli and writes the perceived down vector, otolith AFR, and canal AFR to a data file.

B1.3 **Program Description.** The file Uinface.cpp defines the class TUinface, derived from the Turbo Vision class TApplication. The Borland Turbo Vision users manual should be consulted for a description of the class hierarchy and function. The member function GoVestib performs the reading and writing to the data files, and calls the functions otolithResponse and canalResponse to compute the response of the organs to the acceleration stimuli.

B1.4 **Description of Input.** The vectors should all be expressed in the head coordinate frame, with the z-axis pointing from head to toe, x-axis pointing out, and y-axis pointing to the right. The angular velocity should be expressed in degrees/second, angular acceleration should be expressed in degrees/second^2, and specific force, which is the inertial acceleration plus the effect of gravity, should be in units of earth's gravity (G). The data is expected in columns separated by spaces or tabs, and should be in the following format:

Column:	Item:	Units:
1	Time	seconds
2-4	Angular Velocity x,y,z	deg/s
5-7	Angular Acceleration x,y,z	deg/s^2
8-10	Specific Force x,y,z	G

VESTIB will cease computation after the last line of input is read.

B1.5 **Example User Input Session.** To start the program, type VESTIB at the DOS prompt. The program has menus for selecting the input file name and output file name output file. Select run from the menu to conduct the analysis.

B1.6 **Program Operation.** If the data file is of the correct format, the program will begin computing the vestibular response at each time step. It is possible for the user to crash the program with a bad input data file. The most common error is an error in input file format. The user should ensure that the input is in the correct column format. Check the file format if a read error develops.

B1.7 **Output.** The program writes a data file of time in seconds and perceived down, otolith AFR, and canal AFR components separated by spaces.

B1.8 **Source Program Listing.**

```
// ----------------------------------------------------------
// Program:             uinface.cpp
// Project:       Short Arm Centrifuge, SBIR Phase I
// Date Created:  Aug. 12, 1993
//
// Notes:               This is a Turbo Vision user interface
//                      for the vestibular response subroutines,
//                      (ie. vestib.cpp, otolith.cpp, canal.cpp)
// ----------------------------------------------------------

// define compiler Turbo Vision variables.
#define Uses_TApplication
#define Uses_TEventQueue
#define Uses_TEvent
#define Uses_TApplication
#define Uses_TKeys
#define Uses_TRect
#define Uses_TMenu
#define Uses_TMenuBox
#define Uses_TMenuBar
#define Uses_TMenuItem
#define Uses_TSubMenu
#define Uses_TStatusLine
#define Uses_TStatusDef
#define Uses_TStatusItem
#define Uses_TStaticText
#define Uses_TSItem
#define Uses_TRadioButtons
#define Uses_TDeskTop
#define Uses_TDialog
#define Uses_TButton
#define Uses_TObject
#define Uses_TFileDialog
```

```
#define Uses_TInputLine
#define Uses_TLabel
#define Uses_TButton

// include Borland and custom header files
#include <tv.h>
#include <stdio.h>
#include <stdlib.h>
#include <fstream.h>
#include <string.h>
#include <iomanip.h>
#include <process.h>

//assign command values
const int cmGetInputFile = 1000;
const int cmGetOutputFile = 1001;
const int cmRunSimulation = 1003;
const int cmTimeStepDialog = 1004;
const int cmOverWriteDialog = 1005;

//Global variables
ifstream infile;
ofstream outfile;

//Make a DeltaT structure for retrieval from dialog box
struct DeltaT
{
char dt[10];
};

DeltaT *dtime;

double dt; //time step -- will be string copied from dtime

class TUinface : public TApplication
{
public:
TUinface();
static TDeskTop* initDeskTop(TRect r);
static TMenuBar* initMenuBar(TRect r);
static TStatusLine* initStatusLine(TRect r);
virtual void handleEvent(TEvent& event);
int OverWriteDialog(char fileName[MAXPATH]);
void GetInputFile();
void GetOutputFile();
void GoVestib();
void GetTimeStep();
void SetDefaultTime();
void SetTime();
};
```

```
#include "vestib.cpp"
#include "canal.cpp"
#include "otolith.cpp"

TUinface::TUinface() :
  TProgInit( &TUinface::initStatusLine,
    &TUinface::initMenuBar,
    &TUinface::initDeskTop
    )

{
SetDefaultTime();
}

void TUinface::SetDefaultTime()
{
strcpy(dtime->dt, "0.");
}


int TUinface::OverWriteDialog(char fileName[MAXPATH])
{
   char buff[80];
   int value;
   strcpy(buff, fileName);
   strcat(buff, " exists!");
   int k=strlen(buff)/2 + 10;
   TDialog* d= new TDialog(TRect(40-k, 5, 40+k, 13),
                              buff);

   d->insert(new TButton(TRect(k-12, 4, k-2, 6),
                              "~Y~es", cmYes, bfNormal));

   TButton* button=new TButton(TRect(k+2, 4, k+12, 6),
                              "~N~o", cmNo, bfDefault);
   d->insert(button);

   TView* b = new TInputLine(TRect(k+13, 2, k+13, 3), 0);
   d->insert(b);

   d->insert(new TLabel(TRect(k-5, 2, k+12, 3),
                  "Overwrite?", b));

   button->select();

   ushort control = deskTop->execView(d);
   if (control == cmYes) value = 1;
   else value = 0;

   destroy(d);
   return value;
```

```
}


TDeskTop* TUinface::initDeskTop(TRect r)
{
  // Overrides default desktop to include a status bar at the bottom.
  r.a.y++;
  r.b.y--;
  return new TDeskTop(r);
}



TStatusLine* TUinface::initStatusLine(TRect r)
{
  // This function creates the status bar for the application.
  r.a.y = r.b.y - 1;
  return new TStatusLine(r,
    *new TStatusDef(0, 0xFFFF) +
    *new TStatusItem("~Alt-X~ Exit", kbAltX, cmQuit)
    );
}
void TUinface::GetInputFile()
{
  // This function opens a file dialog box and opens a selected file
  TFileDialog* d = (TFileDialog*) validView (
    new TFileDialog("*.*", "Open", " Input file:", fdOpenButton, 100));
  char fileName[MAXPATH];

  deskTop->execView(d);
  d->getFileName(fileName);                          // get file name
  infile.open(fileName);

  destroy(d);
}

void TUinface::GetOutputFile() //procedure for getting output file name
{
  // This function opens a file dialog box and opens a selected file
  int value = 0;
  while (!value)
    {
    TFileDialog* d = (TFileDialog*) validView (
      new TFileDialog("*.*", "Open", " Output file: ", fdOpenButton, 100));
    char fileName[MAXPATH];

    deskTop->execView(d);
    d->getFileName(fileName);                        // get file name
    if (access(fileName, 0) == 0)            // if file exists
      {
      value = OverWriteDialog(fileName);
    if (value) outfile.open(fileName);
    }
   else
    {
    value = 1;
    outfile.open(fileName);
    }

   destroy(d);
   }

}


void TUinface::GetTimeStep() //procedure for getting time step
{
  TDialog *pd = new TDialog( TRect( 15, 5, 55, 18), "Enter Time Step");
  if (pd)
    {
     pd->insert( new TButton( TRect( 15, 10, 25, 12), "~O~k", cmOK,
                               bfDefault ));
     pd->insert( new TButton( TRect( 28, 10, 38, 12), "~C~ancel",
                               cmCancel, bfNormal));
     TView *b = new TInputLine( TRect(15, 6, 25, 7), 10);
     pd->insert(b);
     pd->insert( new TLabel( TRect(8, 4, 32, 5), "Enter Time Step (sec): ", b));
     pd->setData(dtime);

     ushort control = deskTop->execView(pd);
     if (control != cmCancel)
         {
         pd->getData(dtime);
         SetTime();
         }
    }

 destroy(pd);
}

void TUinface::SetTime()
// get time step from dialog box and convert string
{
dt = atof((char*) dtime->dt);
}

void TUinface::handleEvent(TEvent& event)
{
 // handle event routine.  Responds to main menu selections.
 ushort newMode;
 TApplication::handleEvent(event);
```

```
  if (event.what != evCommand) return;

  switch (event.message.command)
    {
    case cmGetInputFile:            // Open a file for input
      GetInputFile();
      break;
    case cmGetOutputFile:           // Open a file for output
      GetOutputFile();
      break;
    case cmTimeStepDialog:        // Open a dialog box for time step
      GetTimeStep();
      break;
    case cmRunSimulation:
      GoVestib();
    default:
      break;
    }

  clearEvent(event);
}

TMenuBar* TUinface::initMenuBar(TRect r)
{
   r.b.y = r.a.y + 1;

   return new TMenuBar(r,
      *new TSubMenu("~F~ile", kbAltF) +
          *new TMenuItem("~I~nput file", cmGetInputFile,
              kbF3, hcNoContext, "F3") +
          *new TMenuItem("~O~utput file", cmGetOutputFile,
              kbF4, hcNoContext, "F4") +
          *new TMenuItem("~T~ime step", cmTimeStepDialog,
              kbF5, hcNoContext, "F5") +
          *new TMenuItem("~R~un simulation", cmRunSimulation,
              kbF6, hcNoContext, "F6"));

}

int main()
{
  TUinface UserInterface;
  UserInterface.run();
  return 0;
}

// ------------------------------------------------------------
// Program:                    vestib.cpp
// Project:        Short Arm Centrifuge, SBIR Phase I
// Date Created:  Aug. 12, 1993
//
```

```
// Notes:                    This is the main vestibular response function
//                           that calls the otolith and canal functions.
// ----------------------------------------------------------------------


#include <fstream.h>
#include <io.h>
#include <iomanip.h>
#include "vector.h"
#include "matrix.h"

Vector otolithResponse(double, Vector, Vector&);
Vector canalResponse(double, Vector);

void TUinface::GoVestib()
{
  Vector sf, alpha, omega, canAfr, otoAfr, down;
  double time;
  double a,b,c,d,e,f,g,h,i;

  outfile.precision(4);
  outfile.setf(ios::scientific,ios::floatfield);
  outfile.setf(ios::left,ios::adjustfield);

  int count = 0;
  int loop = 0;

  while (!infile.eof())
    {
    count++;
    infile >> time >> a >> b >> c >> d >> e >> f >> g >> h >> i;
    omega.set(a,b,c);
    alpha.set(d,e,f);
    sf.set(g,h,i);
    if (count == 1)
      while (loop < 60.0/dt)
          {
          loop++;
          alpha.set(0,0,0);
          otoAfr = otolithResponse(dt, sf, down);
          canAfr = canalResponse(dt, alpha);
          }
    else
      {
      otoAfr = otolithResponse(dt, sf, down);
      canAfr = canalResponse(dt, alpha);
      }

    outfile << setw(12) << time
            << setw(12) << down.x() << setw(12) << down.y() << setw(12) << down.z()
            << setw(12) << otoAfr.x() << setw(12) << otoAfr.y() << setw(12) << otoAfr.z()
          << setw(12) << canAfr.x() << setw(12) << canAfr.y() << setw(12) << canAfr.z()
          << endl;

   }
}

// ------------------------------------------------------------------
// Program:                otolith.cpp
// Project:        Short Arm Centrifuge, SBIR Phase I
// Date Created:  Aug. 12, 1993
//
// Notes:                  This is the otolith response function.
// ------------------------------------------------------------------

#include "vector.h"
#include "matrix.h"
#include <math.h>

Vector otolithResponse(double dt, Vector sf, Vector &down)
{

/*------------------------------------------------------------------
 Variable and parameter definitions

 dt = time step (sec)
 sf = specific force vector in head coordinates (G)
 sfo = specific force vector in otolith coordinates
 theta = pitch angle of otoliths (assumed 25 degrees)
 RM_ho = rotation matrix from head to otolith coordinates
 gain = otolith dynamics gain constant
 tca = specific force transducer time constant (sec)
 tcb = afferent processing time constant (sec)
 afr = afferent firing rate vector (impulses/sec)
 down = down estimate vector

 Model and parameter values taken from:
   Forsstrom KS, Doty J, Cardullo FM.  Using Human Motion Perception
   Models to Optimize Flight Simulator Algorithms (AIAA No. 85-1743)

 Saccular non-linearity included as described in:
   Borah J, Young LR, Curry RE.  Sensory Mechanism Modelling.  AFHRL-TR-77-70.
   USAF Human Resources Laboratory (Oct 1977).

 The saccular non-linearity is applied to the specific force input after
 passing through the dynamics transfer function.  Specific force is equal to
 the inertial acceleration of the otolith in otolith coordinates.

 No threshold was included because of the non-continuous output.
------------------------------------------------------------------*/

  Vector afr, sfo, afr_h, afr_p;
  static Vector sfo1, afr1;
  Matrix RM_oh, RM_ho;
  const double rd = 57.296;
  const double theta = 25.0/rd;
  const double gain = 9.0, tca = 5.0, tcb = 10.0;

  RM_ho = setRMatrix(0, theta, 0);
  RM_oh = trans(RM_ho);                              // transpose of original matrix

// rotate specific force input to otolith coordinate frame
  sfo = RM_ho*sf;

// define constants for otolith response to specific force input
  double a0 = dt + 2.0*tca;
  double a1 = dt - 2.0*tca;
  double b0 = gain*tca*(dt + 2.0*tcb);
  double b1 = gain*tca*(dt - 2.0*tcb);
  afr = (b1*sfo1 + b0*sfo - a1*afr1)/a0;

// save variables for next call
  sfo1 = sfo;
  afr1 = afr;

// apply observed non-linearity of saccular otolith to afferent firing
// rate in otolith coordinate frame to create perceived afr. Perceived
// afr permits computation of perceived down vector.
  afr_p = afr;
  if (afr.z() > -45.0*cos(theta))
    afr_p.set_z(-0.4*45*cos(theta) + 0.6*afr.z());
  else
    afr_p.set_z(-45*cos(theta));

// compute the direction of the down estimate by computing its normalized
// component vectors (in otolith coordinates). Rotate the down estimate
// back to head coordinates.
  if (mr,g(afr_p) > 0.0)
    down = -1.0*RM_oh*(afr_p/mag(afr_p));            // unit afr vector
  else
    down.set(0,0,0);

// rotate the afferent firing rates back to the head frame.
  afr_h = RM_oh*afr;

//  return afr_h;
  return afr_h;
}

// ---------------------------------------------------------------
// Program:                 canal.cpp
// Project:         Short Arm Centrifuge, SBIR Phase I
// Date Created:  Aug. 12, 1993
```

```
//
// Notes:                    This is the canal response function.
// ------------------------------------------------------------


#include "vector.h"
#include "matrix.h"
#include <math.h>

Vector canalResponse(double dt, Vector alpha)
{

/*------------------------------------------------------------
 Variable and parameter definitions

 dt = time step (sec)
 alpha = angular acceleration (degs/sec^2)
 tlo = canal long time constant vector
 tad = canal adaptation time constant vector
 tle = canal lead operator time constant vector
 afr = afferent firing rate vector (impulses/sec)

 Angular acceleration in the head coordinate system.  The semicircular canals
 are approximated to be in the head coordinate system.

 No threshold was included because of the non-continuous output.  Instead,
 we implemented the vestibular threshold unit concept.

------------------------------------------------------------*/

  Vector asum, bsum, afr;
  Vector tlo;
  Vector a0, a1, a2, b0, b1, b2;
  static Vector afr1, afr2, alpha1, alpha2;
  const double tad = 30.0, tle = 0.01;
  int i, k;

  tlo.set(6.1, 5.3, 10.2);

// calculate the numerator and denominator of the weighting constants
//
// b --> numerator coefficients, a --> denominator coefficients
//
  Vector dt_temp(dt,dt,dt);           // vectorize quantity for vector operations
  b0 = 2.0*tad*(dt + 2.0*tle)*tlo;
  b1 = -8.0*tad*tle*tlo;
  b2 = 2.0*tad*(2.0*tle - dt)*tlo;
  a0 = (dt + 2.0*tad)*(dt_temp + 2.0*tlo);
  a1 = 2.0*dt*dt_temp - 8.0*tad*tlo;
  a2 = (dt - 2.0*tad)*(dt_temp - 2.0*tlo);
// compute afferent firing rate components

  asum.set_x(a1.x()*afr1.x() + a2.x()*afr2.x());
  asum.set_y(a1.y()*afr1.y() + a2.y()*afr2.y());
  asum.set_z(a1.z()*afr1.z() + a2.z()*afr2.z());
  bsum.set_x(b0.x()*alpha.x() + b1.x()*alpha1.x() + b2.x()*alpha2.x());
  bsum.set_y(b0.y()*alpha.y() + b1.y()*alpha1.y() + b2.y()*alpha2.y());
  bsum.set_z(b0.z()*alpha.z() + b1.z()*alpha1.z() + b2.z()*alpha2.z());
  afr.set_x((bsum.x() - asum.x())/a0.x());
  afr.set_y((bsum.y() - asum.y())/a0.y());
  afr.set_z((bsum.z() - asum.z())/a0.z());

// save old values
  afr2 = afr1;
  afr1 = afr;
  alpha2 = alpha1;
  alpha1 = alpha;

  return afr;
}
```

APPENDIX B2
Centrifuge and Shuttle Simulation Program

B2.0 **Program Name.** ACCELS.EXE for MS-DOS.

B2.1 **Purpose.** ACCELS simulates the dynamics of a short radius centrifuge mounted on the space shuttle. The output can consist of linear or angular positions, velocities, and accelerations, or dynamic forces and moments acting on the centrifuge or shuttle. The user is able to select the size, position, number, and gender of the centrifuge riders, as well as input simulation parameters and shuttle and centrifuge dimensions and mass properties.

B2.2 **General Description of Program Flow.** On execution, the user is presented with an interface for selecting simulation, shuttle, centrifuge, rider parameters, and the output file name and output variables. The program can be run with default values, however at least one rider must be selected. When the selections have been made, activating the **Simulate** button initiates the simulation.

B2.3 **Program Description.** The file Accels.cpp defines the class Centrifuge, derived from the Turbo Vision class TApplication. The Borland Turbo Vision users manual should be consulted for a description of the class hierarchy and function. Function startSim() performs the actual simulation and data recording to the output file.

B2.4 **Description of Input.** There is no input file required for ACCELS. The user should enter the output file name, output file variables, shuttle mass and inertia values, simulation parameters, rider positions and description, and centrifuge inertia and dimensions into the appropriate input boxes. The values in each input box can be reset to the default values with the **Default** buttons. The simulation can be initiated with the **Simulate** button.

B2.5 **Example User Input Session.** To start the program, type ACCELS at the DOS prompt. Using the mouse or keyboard Alt-F keys, select the **File** menu selection. Select **New** from the submenus, to open a new simulation worksheet. Now, individually select the parameters of the simulation, with the following directions:

- **Select output file name:** Using the mouse or keyboard, select the name of the output file to store the data. If the file is a new one, simply type in the name of the file. If the file exists, the program will ask you to confirm whether you want to replace the file.

- **Shuttle mass and inertia:** The default shuttle mass and inertia matrix can be changed if desired.

- **Simulation time and acceleration:** The duration of the simulation is specified by the first two values. The centrifuge is initially at rest. The duration of the first phase represents the time to accelerate to a constant G and remain at that acceleration. After the first phase of the simulation, the centrifuge will decelerate to rest and remain at rest until the total simulation time has elapsed. This feature allows the long term effects of the centrifuge motion to be assessed. The time interval can be changed if desired. The maximum acceleration for the simulation represents the acceleration in G's at the location of subject #1 center of mass. The acceleration onset and tangential acceleration limit for the centrifuge can also be adjusted.

- **Rider parameters:** The occupancy, percentile, gender, and head position of up to four riders can be specified. The first position must be occupied in order for the program to work properly. Other riders can be added as desired, with position 3 being opposite position 1, and position 2 opposite position 4. The program will automatically use the appropriate mass, inertia, and center of mass values for the riders according to the gender and percentile selected. The position of the *rider is considered to be from the center of the centrifuge to the top of* the head.

- **Centrifuge inertia and dimensions:** The inertia of the centrifuge itself, and the position of the center of the centrifuge with respect to the shuttle center of mass can be edited in this dialog box.

- **Output Variables:** Check the boxes of the variables to be written to the output file. The variables are all vectors, and all three components of each vector will be saved. Time is automatically recorded in the first column of the output file.

Once all the values have been selected (remember to specify at least the first crewmember position as occupied and to select an output file name), the **Simulate** button can be selected to run the simulation.

B2.6 **Program Operation.** When the Simulate button is activated, ACCELS will *perform the simulation* and print results to the data file. When ACCELS is finished, a dialog box will appear announcing the completion. If the program does not operate, the most likely causes for the error is that the crewmember position #1 was not set to occupied, or no output file name was selected.

B2.7 **Output.** The program writes a data file of time in seconds and the selected output variables to the output file.

B2.8 Source Program Listing.

```
// -----------------------------------------------------------
// Program:           Accels.cpp
// Project:     Short Arm Centrifuge, SBIR Phase I
// Date Created: May 21, 1993
//
// Notes:             This program uses a Turbo Vision interface and
//                    computes the kinematics and dynamics of a short
//                    radius centrifuge operating on the space shuttle.
// -----------------------------------------------------------


// define Turbo Vision compiler variables.
#define Uses_TEventQueue
#define Uses_TEvent
#define Uses_TApplication
#define Uses_TKeys
#define Uses_TRect
#define Uses_TMenu
#define Uses_TMenuBox
#define Uses_TMenuBar
#define Uses_TMenuItem
#define Uses_TSubMenu
#define Uses_TStatusLine
#define Uses_TStatusDef
#define Uses_TStatusItem
#define Uses_TStaticText
#define Uses_TSItem
#define Uses_TRadioButtons
#define Uses_TCheckBoxes
#define Uses_TDeskTop
#define Uses_TDialog
#define Uses_TButton
#define Uses_TObject
#define Uses_TFileDialog
#define Uses_TInputLine
#define Uses_TLabel

// include Borland and custom header files
#include <tv.h>
#include "matrix.h"
#include "vector.h"
#include <stdio.h>
#include <stdlib.h>
#include <fstream.h>
#include <string.h>
#include <iomanip.h>
```

```
// include other code

// assign command values, use values over 1000 to always be visible to Turbo Vision
const int cmNew = 1000;
const int cmHelpAbout = 1001;
const int cmStartSim = 1002;
const int cmRefresh = 1003;
const int cmFile = 1004;
const int cmShuttle = 1005;
const int cmTime = 1006;
const int cmSubject = 1007;
const int cmCentrifuge = 1008;
const int cmOutputVars = 1009;

//------------------------ global variables ------------------------

ofstream f1;                              // output file

struct SimData                            // simulation data from dialog boxes
{
  char endTime1[10];                      // end of first simulation phase
  char endTime2[10];                      // end of total simulation
  char dt[10];
  char maxG[10];
  char onsetG[10];
  char maxTanG[10];
} simData;

struct ShuttleData                        // shuttle data from dialog boxes
{
  char mass[10];
  char Ixx[10];  char Ixy[10];  char Ixz[10];
  char Iyx[10];  char Iyy[10];  char Iyz[10];
  char Izx[10];  char Izy[10];  char Izz[10];
} shuttleData;

struct SubjectData                        // subject data from dialog boxes
{
  struct Subject
    {
    ushort occupied;
    ushort percent;
    ushort gender;
    char position[10];
    } s[4];
} subjectData;

struct CfugeData                                      // short arm centrifuge from dialog boxes
{
  char I[10];
  char com[10];
```

```
} cfugeData;

struct OutputData                         // variables to output file
{
  ushort checkBox;
} outputData;


double endTime1;                          // end simulation phase 1
double endTime2;                          // end total simulation
double dt;                                // simulation time step
double maxG;                              // maximum acceleration in G at subject #1 c.o.m.
double onsetG;                            // acceleration onset in G/sec
double maxTanG;                           // maximum tangential acceleration in G

struct Shuttle
{
  double mass;
  Matrix I;
} shuttle;

struct Person
{
  ushort occupied;                        // 0 if person not on centrifuge, 1 if he is
  double mass;
  double com;                             // center of mass distance from top of head (cm)
  double toh;                             // centrifuge to top of head distance (cm)
  double I;                               // kg-m^2
} p[4];

struct Cfuge                              // centrifuge inertia and center of mass relative
{                                         // to space shuttle center of mass
  double I;
  double com;
} cfuge;


// ------------------------ User interface ------------------------

class Centrifuge : public TApplication
{
  // Define BRC application class Centrifuge.
  private:

  public:
    Centrifuge();
    static TDeskTop* initDeskTop(TRect r);
    static TMenuBar* initMenuBar(TRect r);
    static TStatusLine* initStatusLine(TRect r);
    virtual void handleEvent(TEvent& event);
    void openFile();
    void newSim();
    int  checkParams();
    void startSim();
    void setSimParams();
    void setSimParamsDft();
    void setShuttleParams();
    void setShuttleParamsDft();
    void setSubjectParams();
    void setSubjectParamsDft();
    void setCfugeParams();
    void setCfugeParamsDft();
    void selectOutput();
    void helpAbout();
    void simFinished();
};

Centrifuge::Centrifuge() : TProgInit(&Centrifuge::initStatusLine,
                                     &Centrifuge::initMenuBar,
                                     &Centrifuge::initDeskTop)
{
  // Constructor displays "about" box at startup.
  TEvent event;

  // Set default values.
  setSimParamsDft();
  setShuttleParamsDft();
  setSubjectParamsDft();
  setCfugeParamsDft();

  // Display "about" box.
  event.what = evCommand;
  event.message.command = cmHelpAbout;
  putEvent(event);
}

void Centrifuge::helpAbout()
{
  // This function displays startup info.
  TDialog* d = (TDialog*) validView(
    new TDialog(TRect(20,5,60,15), "Short Arm Centrifuge"));

  if (!d) return;
  d->insert(new TStaticText(TRect(1,1,39,7),
           "\n\003Short Arm Centrifuge Software"
           "\n\003Biodynamic Research Corporation"
           "\n\003November, 1993"));
  d->insert(new TButton(TRect(14,7,26,9), "OK", cmOK, bfDefault));
  deskTop->execView(d);
  destroy(d);
}
```

```
void Centrifuge::simFinished()
{
  // This function displays a simulation finished message.
  TDialog* d = (TDialog*) validView(
    new TDialog(TRect(20,5,60,15), "Message Box"));

  if (!d) return;
  d->insert(new TStaticText(TRect(1,1,39,7 ),
              "\n\003The Simulation is Complete"));
  d->insert(new TButton(TRect(14,7,26,9), "OK", cmOK, bfDefault));
  deskTop->execView(d);
  destroy(d);
}

void Centrifuge::handleEvent(TEvent& event)
{
  // Centrifuge handle event routine.  Responds to main menu selections.
  ushort newMode;
  TApplication::handleEvent(event);

  if (event.what != evCommand) return;

  switch (event.message.command)
    {
    case cmNew:                              // Open new simulation sheet
      newSim();
      break;
    case cmHelpAbout:                        // Display help about box
      helpAbout();
      break;
    default:
      break;
    }

  clearEvent(event);
}

TMenuBar* Centrifuge::initMenuBar(TRect r)
{
  // This function creates the MenuBar for the application.
  r.b.y = r.a.y + 1;
  return new TMenuBar(r,
    *new TSubMenu("~F~ile", kbAltF) +
      *new TMenuItem("~N~ew", cmNew, kbAltN, hcNoContext, "Alt-N") +
      newLine() +
      *new TMenuItem("E~x~it", cmQuit, kbAltX, hcNoContext, "Alt-X")
    );
}

TDeskTop* Centrifuge::initDeskTop(TRect r)
{
  // Overrides default desktop to include a status bar at the bottom.
  r.a.y++;
  r.b.y–;
  return new TDeskTop(r);
}

TStatusLine* Centrifuge::initStatusLine(TRect r)
{
  // This function creates the status bar for the application.
  r.a.y = r.b.y - 1;
  return new TStatusLine(r,
    *new TStatusDef(0, 0xFFFF) +
    *new TStatusItem("~Alt-X~ Exit", kbAltX, cmQuit)
    );
}

class TNewDialog : public TDialog
{
  // Derived class for a new dialog box that can open other dialog boxes.
  public:
    TNewDialog(TRect&, char*);                    // new constructor
    void handleEvent(TEvent&);                    // new handleEvent routine
    virtual Boolean valid(ushort);
};

TNewDialog::TNewDialog(TRect& r, char* title) :
              TDialog(r, title),
              TWindowInit(&TNewDialog::initFrame)
{  // Constructor derived from TDialog constructor.
}

void TNewDialog::handleEvent(TEvent& event)
{
  // New dialog box handleEvent routine.
  TDialog::handleEvent(event);                    // derived from TDialog handleEvent

  if (event.what == evKeyDown)
    if (event.keyDown.keyCode == kbEsc)                 // if Esc, cancel from dialog bo
      {
      event.what = evCommand;
      event.message.command = cmCancel;
      putEvent(event);
      }

  if (event.what == evCommand)
    endModal(event.message.command);              // end modal state and set command

  clearEvent(event);
}

Boolean TNewDialog::valid(ushort command)
```

```
{
  // New dialog box valid routine - returns endModal command to calling function.
  if (command == cmCancel)
    return True;
  else
    return TGroup::valid(command);                    // return command value to calling function
}

void Centrifuge::newSim(void)
{
  // This function displays a worksheet for initializing a sac simulation.
  TView* b;

  TNewDialog* pd = new TNewDialog (TRect(1,1,75,20), "Short Arm Centrifuge Simulation");
  if (pd)
    {
    pd->insert(new TButton (TRect(50,16,60,18),"~S~imulate", cmStartSim, bfNormal));
    pd->insert(new TButton (TRect(63,16,73,18),"~C~ancel", cmCancel, bfDefault));

    pd->insert(new TStaticText(TRect(2,2,40,3),"Select output file name:"));
    pd->insert(new TButton (TRect(50,2,60,4),"Select", cmFile, bfNormal));

    pd->insert(new TStaticText(TRect(2,4,40,5),"Shuttle mass and inertia:"));
    pd->insert(new TButton (TRect(50,4,60,6),"Select", cmShuttle, bfNormal));

    pd->insert(new TStaticText(TRect(2,6,40,7),"Simulation time and acceleration:"));
    pd->insert(new TButton (TRect(50,6,60,8),"Select", cmTime, bfNormal));

    pd->insert(new TStaticText(TRect(2,8,40,9),"Rider parameters:"));
    pd->insert(new TButton (TRect(50,8,60,10),"Select", cmSubject, bfNormal));

    pd->insert(new TStaticText(TRect(2,10,40,11),"Centrifuge inertia and dimensions:"));
    pd->insert(new TButton (TRect(50,10,60,12),"Select", cmCentrifuge, bfNormal));

    pd->insert(new TStaticText(TRect(2,12,40,13),"Output Variables"));
    pd->insert(new TButton (TRect(50,12,60,14),"Select", cmOutputVars, bfNormal));

    // allow user to select different parameter worksheets
    ushort control;
    while (control != cmCancel)
      {
      control = deskTop->execView(pd);
      switch (control)
          {
          case cmStartSim:
            int error = checkParams();
            if (!error) startSim();
            simFinished();
            control = cmCancel;
            break;
          case cmShuttle:
            setShuttleParams();
            break;
          case cmTime:
            setSimParams();
            break;
          case cmSubject:
            setSubjectParams();
            break;
          case cmCentrifuge:
            setCfugeParams();
            break;
          case cmFile:
            openFile();
            break;
          case cmOutputVars:
            selectOutput();
            break;
          default:
            break;
          }
      }
    }
  destroy(pd);
}

int Centrifuge::checkParams()
{
  // This function checks the input parameters to make sure there are
  // no obvious errors and to convert character strings to numbers.
  Vector a, b, c;     // temporary variables to assign matrix elements
  int flag = 0;                  // error flag

  dt = atof((char*) simData.dt);
  endTime1 = atof((char*) simData.endTime1);
  endTime2 = atof((char*) simData.endTime2);
  maxG = atof((char*) simData.maxG);
  onsetG = atof((char*) simData.onsetG);
  maxTanG = atof((char*) simData.maxTanG);
  if (dt <= 0.0 || endTime1 <= 0.0 || endTime2 <= 0.0 || maxG <= 0.0 ||
      onsetG <= 0.0 || maxTanG <= 0.0) flag = 1;

  shuttle.mass = atof((char*) shuttleData.mass);
  if (shuttle.mass <= 0.0) flag = 1;

  a.set_x(atof((char*) shuttleData.Ixx));
  a.set_y(atof((char*) shuttleData.Ixy));
  a.set_z(atof((char*) shuttleData.Ixz));
  b.set_x(atof((char*) shuttleData.Iyx));
  b.set_y(atof((char*) shuttleData.Iyy));
  b.set_z(atof((char*) shuttleData.Iyz));
  c.set_x(atof((char*) shuttleData.Izx));
```

```
c.set_y(atof((char*) shuttleData.Izy));
c.set_z(atof((char*) shuttleData.Izz));
shuttle.I.set(a,b,c);                    // set moment of inertia matrix

if (a.x() <= 0.0 || b.y() <= 0.0 || c.z() <= 0.0) flag = 1;  // principal m.o.i.
if (a.y() != b.x() || a.z() != c.x() || b.z() != c.y()) flag = 1;  // product symmetry

for (int i=0; i<4; i++)
  {
  p[i].occupied = subjectData.s[i].occupied;
  p[i].toh = atof((char*) subjectData.s[i].position);
  if (subjectData.s[i].percent == 0)
    if (subjectData.s[i].gender == 0)                     // if male
        {
        p[i].mass = 65.8;                                 // kg
        p[i].com = 51.6;                                  // cm from head
        p[i].I = 3.8;                                     // kg-m^2
        }
    else                                                  // if female
        {
        p[i].mass = 41.0;
        p[i].com = 45.2;
        p[i].I = 2.4;
        }
  else if (subjectData.s[i].percent == 1)
    if (subjectData.s[i].gender == 0)
        {
        p[i].mass = 82.2;
        p[i].com = 53.5;
        p[i].I = 4.9;
        }
    else
        {
        p[i].mass = 51.5;
        p[i].com = 46.7;
        p[i].I = 3.1;
        }
  else
    if (subjectData.s[i].gender == 0)
        {
        p[i].mass = 98.5;
        p[i].com = 55.4;
        p[i].I = 6.0;
        }
    else
        {
        p[i].mass = 61.7;
        p[i].com = 48.0;
        p[i].I = 3.8;
        }
  }
  cfuge.I = atof((char*) cfugeData.I);
  cfuge.com = -atof((char*) cfugeData.com);          // negative because of shuttle axes
  if (cfuge.I <= 0.0) flag = 1;

  return flag;                                       // return error flag value
}

void Centrifuge::setSimParams()
{
  // This function displays a worksheet for initializing the simulation parameters.
  TView* b;

  TNewDialog* pd = new TNewDialog (TRect(1,1,75,20), "Simulation Parameters");
  if (pd)
    {
    pd->insert(new TButton (TRect(37,16,47,18),"~O~K", cmOK, bfDefault));
    pd->insert(new TButton (TRect(50,16,60,18),"~D~efault", cmRefresh, bfNormal));
    pd->insert(new TButton (TRect(63,16,73,18),"~C~ancel", cmCancel, bfNormal));

    b = new TInputLine (TRect(50,2,60,3), 10);
    pd->insert(b);
    pd->insert(new TLabel (TRect(2,2,40,3),"Duration of first phase (minutes):", b));

    b = new TInputLine (TRect(50,4,60,5), 10);
    pd->insert(b);
    pd->insert(new TLabel (TRect(2,4,49,5),"Duration of total simulation (minutes):", b));

    b = new TInputLine (TRect(50,6,60,7), 10);
    pd->insert(b);
    pd->insert(new TLabel (TRect(2,6,40,7),"Time step interval (seconds):", b));

    b = new TInputLine (TRect(50,8,60,9), 10);
    pd->insert(b);
    pd->insert(new TLabel (TRect(2,8,49,9),"Maximum acceleration at subject 1 COM (G):", b));

    b = new TInputLine (TRect(50,10,60,11), 10);
    pd->insert(b);
    pd->insert(new TLabel (TRect(2,10,40,11),"Acceleration onset (G/sec):", b));

    b = new TInputLine (TRect(50,12,60,13), 10);
    pd->insert(b);
    pd->insert(new TLabel (TRect(2,12,49,13),"Tangential acceleration limit (G):", b));

    pd->setData(&simData);
    ushort control;
    while (control != cmCancel && control != cmOK)
      {
      control = deskTop->execView(pd);
      switch (control)
          {
          case cmRefresh:
```

```
          setSimParamsDft();
          pd->setData(&simData);
          break;
        default:
          break;
        }
    }
  if (control == cmOK) pd->getData(&simData);

  }
 destroy(pd);
}

void Centrifuge::setSimParamsDft()
{
 // This function sets the default values for the simulation data.
 strcpy(simData.endTime1, "1");                    // minutes
 strcpy(simData.endTime2, "2");                    // minutes
 strcpy(simData.dt, "0.025");                      // seconds
 strcpy(simData.maxG, "2");                        // G
 strcpy(simData.onsetG, "0.01");                            // G/sec
 strcpy(simData.maxTanG, "0.05");                  // G
}

void Centrifuge::setShuttleParams()
{
 // This function displays a worksheet for initializing the shuttle parameters.
 TView* b;

 TNewDialog* pd = new TNewDialog (TRect(1,1,75,20), "Shuttle Parameters");
 if (pd)
  {
  pd->insert(new TButton (TRect(37,16,47,18),"~O~K", cmOK, bfDefault));
  pd->insert(new TButton (TRect(50,16,60,18),"~D~efault", cmRefresh, bfNormal));
  pd->insert(new TButton (TRect(63,16,73,18),"~C~ancel", cmCancel, bfNormal));

  pd->insert(new TInputLine (TRect(30,3,40,4), 10));
  pd->insert(new TStaticText (TRect(2,3,20,4),"Shuttle mass (kg):"));

  pd->insert(new TStaticText (TRect(2,5,30,6),"Shuttle inertias (kg-m^2):"));
  pd->insert(new TInputLine (TRect(30,6,41,7), 10));
  pd->insert(new TInputLine (TRect(42,6,53,7), 10));
  pd->insert(new TInputLine (TRect(54,6,65,7), 10));
  pd->insert(new TStaticText (TRect(2,6,20,7),"Ixx, Ixy, Ixz:"));
  pd->insert(new TInputLine (TRect(30,7,41,8), 10));
  pd->insert(new TInputLine (TRect(42,7,53,8), 10));
  pd->insert(new TInputLine (TRect(54,7,65,8), 10));
  pd->insert(new TStaticText (TRect(2,7,20,8),"Iyx, Iyy, Iyz:"));
  pd->insert(new TInputLine (TRect(30,8,41,9), 10));
  pd->insert(new TInputLine (TRect(42,8,53,9), 10));
  pd->insert(new TInputLine (TRect(54,8,65,9), 10));
  pd->insert(new TStaticText (TRect(2,8,20,9),"Izx, Izy, Izz:"));

  pd->setData(&shuttleData);
  ushort control;
  while (control != cmCancel && control != cmOK)
   {
   control = deskTop->execView(pd);
   switch (control)
     {
     case cmRefresh:
      setShuttleParamsDft();
      pd->setData(&shuttleData);
      break;
     default:
      break;
     }
   }
  if (control == cmOK) pd->getData(&shuttleData);

  }
 destroy(pd);
}

void Centrifuge::setShuttleParamsDft()
{
 // This function sets the default values for the shuttle data.
 strcpy(shuttleData.mass, "98122");                  // shuttle mass in kg
 strcpy(shuttleData.Ixx, "1453255");                 // shuttle inertias in kg-m^2
 strcpy(shuttleData.Ixy, "-7544");
 strcpy(shuttleData.Ixz, "-335642");
 strcpy(shuttleData.Iyx, "-7544");
 strcpy(shuttleData.Iyy, "10077164");
 strcpy(shuttleData.Iyz, "-10259");
 strcpy(shuttleData.Izx, "-335642");
 strcpy(shuttleData.Izy, "-10259");
 strcpy(shuttleData.Izz, "10346577");
}

void Centrifuge::setSubjectParams()
{
 // This function displays a worksheet for initializing the subject parameters.
 TView* b;

 TNewDialog* pd = new TNewDialog (TRect(1,0,76,22), "Rider Parameters");
 if (pd)
  {
  pd->insert(new TButton (TRect(37,19,47,21),"~O~K", cmOK, bfDefault));
  pd->insert(new TButton (TRect(50,19,60,21),"~D~efault", cmRefresh, bfNormal));
  pd->insert(new TButton (TRect(63,19,73,21),"~C~ancel", cmCancel, bfNormal));

  // subject positions and types
```

```
pd->insert(new TStaticText (TRect(3,1,10,2), "Rider"));
pd->insert(new TStaticText (TRect(11,1,26,2), "Occupied?"));
pd->insert(new TStaticText (TRect(27,1,39,2), "Percentile"));
pd->insert(new TStaticText (TRect(40,1,50,2), "M/F ?"));
pd->insert(new TStaticText (TRect(51,1,72,2), "Head position (cm)"));

pd->insert(new TStaticText (TRect(3,3,12,4), "#1"));
b = new TRadioButtons (TRect(11,3,26,5),
      new TSItem("Vacant", new TSItem("Occupied", 0)));
pd->insert(b);

b = new TRadioButtons (TRect(27,3,38,6),
      new TSItem("5%", new TSItem("50%", new TSItem("95%", 0))));
pd->insert(b);

b = new TRadioButtons (TRect(39,3,50,5),
      new TSItem("M", new TSItem("F", 0)));
pd->insert(b);

pd->insert(new TInputLine (TRect(51,3,68,4), 10));

pd->insert(new TStaticText (TRect(3,7,12,8), "#2"));
b = new TRadioButtons (TRect(11,7,26,9),
      new TSItem("Vacant", new TSItem("Occupied", 0)));
pd->insert(b);

b = new TRadioButtons (TRect(27,7,38,10),
      new TSItem("5%", new TSItem("50%", new TSItem("95%", 0))));
pd->insert(b);

b = new TRadioButtons (TRect(39,7,50,9),
      new TSItem("M", new TSItem("F", 0)));
pd->insert(b);

pd->insert(new TInputLine (TRect(51,7,68,8), 10));

pd->insert(new TStaticText (TRect(3,11,12,12), "#3"));
b = new TRadioButtons (TRect(11,11,26,13),
      new TSItem("Vacant", new TSItem("Occupied", 0)));
pd->insert(b);

b = new TRadioButtons (TRect(27,11,38,14),
      new TSItem("5%", new TSItem("50%", new TSItem("95%", 0))));
pd->insert(b);

b = new TRadioButtons (TRect(39,11,50,13),
      new TSItem("M", new TSItem("F", 0)));
pd->insert(b);

pd->insert(new TInputLine (TRect(51,11,68,12), 10));
```

```
    pd->insert(new TStaticText (TRect(3,15,12,16), "#4"));
    b = new TRadioButtons (TRect(11,15,26,17),
         new TSItem("Vacant", new TSItem("Occupied", 0)));
    pd->insert(b);

    b = new TRadioButtons (TRect(27,15,38,18),
         new TSItem("5%", new TSItem("50%", new TSItem("95%", 0))));
    pd->insert(b);

    b = new TRadioButtons (TRect(39,15,50,17),
         new TSItem("M", new TSItem("F", 0)));
    pd->insert(b);

    pd->insert(new TInputLine (TRect(51,15,68,16), 10));

    pd->setData(&subjectData);
    ushort control;
    while (control != cmCancel && control != cmOK)
     {
     control = deskTop->execView(pd);
     switch (control)
         {
         case cmRefresh:
           setSubjectParamsDft();
           pd->setData(&subjectData);
           break;
         default:
           break;
         }
     }
    if (control == cmOK) pd->getData(&subjectData);

    }
  destroy(pd);
}

void Centrifuge::setSubjectParamsDft()
{
  // This function sets the default values for the subject data.
  for (int i=0; i<4; i++)                         // i is subject number
    {
    subjectData.s[i].occupied = 0;                      // default is unoccupied
    subjectData.s[i].percent = 0;                 // default is 5%
    subjectData.s[i].gender = 0;                  // default is male
    strcpy(subjectData.s[i].position, "100");          // default is 100 cm
    }                                             // position is the top of the head
}

void Centrifuge::setCfugeParams()
{
  // This function displays a worksheet for initializing the Centrifuge parameters.
```

```
 TView* b;

 TNewDialog* pd = new TNewDialog (TRect(1,1,75,20), "Centrifuge Parameters");
 if (pd)
   {
   pd->insert(new TButton (TRect(37,16,47,18),"~O~K", cmOK, bfDefault));
   pd->insert(new TButton (TRect(50,16,60,18),"~D~efault", cmRefresh, bfNormal));
   pd->insert(new TButton (TRect(63,16,73,18),"~C~ancel", cmCancel, bfNormal));

   b = new TInputLine (TRect(55,3,65,4), 10);
   pd->insert(b);
   pd->insert(new TLabel (TRect(2,3,45,4),"Centrifuge moment of inertia (kg-m^2):", b));

   b = new TInputLine (TRect(55,5,65,6), 10);
   pd->insert(b);
   pd->insert(new TLabel (TRect(2,5,52,6),"Cfuge COM position relative to shuttle COM (m):", b));

   pd->setData(&cfugeData);
   ushort control;
   while (control != cmCancel && control != cmOK)
     {
     control = deskTop->execView(pd);
     switch (control)
         {
         case cmRefresh:
           setCfugeParamsDft();
           pd->setData(&cfugeData);
           break;
         default:
           break;
         }
     }
   if (control == cmOK) pd->getData(&cfugeData);
   }
  destroy(pd);
}

void Centrifuge::setCfugeParamsDft()
{
  // This function sets the default values for the centrifuge center of mass
  // location (forward of the shuttle com) and inertia (only inertias
  // about the rotating axis are considered).
  strcpy(cfugeData.I, "12.5");                          // made up value, kg-m^2
  strcpy(cfugeData.com, "12.9");                        // from Meeker paper, m
}

void Centrifuge::selectOutput()
{
  // This function displays a worksheet for selecting the output variables.
  TView* b;
  TDialog* pd = new TDialog (TRect(1,2,76,20), "Output Variables");
  if (pd)
    {
    pd->insert(new TButton (TRect(50,15,60,17)," ~O~K", cmOK, bfDefault));
    pd->insert(new TButton (TRect(63,15,73,17)," ~C~ancel", cmCancel, bfNormal));

    // output selections

    pd->insert(new TStaticText (TRect(11,2,40,3), "Time"));

    b = new TCheckBoxes (TRect(11,3,45,12),
          new TSItem("Cfuge Angular Velocity", new TSItem("Cfuge Angular Acceleration",
          new TSItem("Cfuge Drive Moment", new TSItem("SS Force",
          new TSItem("SS Moment",         new TSItem("SS Acceleration",
          new TSItem("SS Angular Acceleration", new TSItem("SS COM Displacement",
          new TSItem("SS Angular Displacement",0)
          ))))))))));
    pd->insert(b);

    pd->insert(new TStaticText (TRect(50,2,63,3), "(sec)"));
    pd->insert(new TStaticText (TRect(50,3,63,4), "(deg/sec)"));
    pd->insert(new TStaticText (TRect(50,4,63,5), "(deg/sec^2)"));
    pd->insert(new TStaticText (TRect(50,5,63,6), "(Nm)"));
    pd->insert(new TStaticText (TRect(50,6,63,7), "(N)"));
    pd->insert(new TStaticText (TRect(50,7,63,8), "(Nm)"));
    pd->insert(new TStaticText (TRect(50,8,63,9), "(G)"));
    pd->insert(new TStaticText (TRect(50,9,63,10), "(deg/sec^2)"));
    pd->insert(new TStaticText (TRect(50,10,63,11), "(m)"));
    pd->insert(new TStaticText (TRect(50,11,63,12), "(deg)"));

    pd->setData(&outputData);
    ushort control = deskTop->execView(pd);
    if (control == cmOK) pd->getData(&outputData);

    }
  destroy(pd);
}

void Centrifuge::openFile()
{
  // This function opens a file dialog box and opens a selected file for output.
  TFileDialog* d = (TFileDialog*) validView (
    new TFileDialog("*.*", "Open a File", "~N~ame", fdOpenButton, 100));
  char fileName[MAXPATH];

  if (d)
    {
    deskTop->execView(d);
    d->getFileName(fileName);                       // get output file name
    f1.open(fileName);                              // open output file
    }
  destroy(d);
}

// ---------------------- Application Main Routine -------------------

int main(void)
{
  // This is the main routine for the application.
  Centrifuge sac;
  sac.run();
  return 0;
}

// ----------------- Centrifuge/Shuttle Simulation routine -------------

void Centrifuge::startSim(void)
{
  // This function initializes variables for the simulation and loops through
  // the specified simulation time.  All quantities expressed in metric units.
  const double g = 9.81;                                // m/s^2
  const double pi = 3.141592654;                             // pi
  const double rd = 180.0/pi;                           // radians to degrees
  double time = 0.0;
  double inertia = 0.0;                                 // temporary variable
  Vector gamma_c, omega_c, alpha_c;                     // angular kinematics of cfuge
  Vector H_c;                                           // angular momentum of cfuge
  Vector omega_cLast, alpha_cLast;                      // last time step values
  Vector r_p[4], v_p[4], a_p[4];                        // subject kinematics
  Vector r_pr[4], r_prtotal[4];                         // rotated subject position
                                                        // with cfuge position

  Vector F_p[4], M_p;                                   // subject dynamics
  Vector r_cfuge;                                            // centrifuge com position
  Vector r_ss, v_ss, a_ss;                                   // shuttle kinematics
  Vector omega_ssLast, alpha_ssLast;                    // last time step values
  Vector gamma_ss, omega_ss, alpha_ss;                  // shuttle angular kinematics
  Vector omega_tot, alpha_tot;                          // subject total angular vel. and accel.
  Vector v_ssLast, a_ssLast;                            // last time step values
  Vector F_ss, M_ss;                                    // shuttle force, moment, ang. momentum
  Vector M_imb, M_cfuge, M_gyro;                        // imbalance,drive,gyroscopic moments
  Vector atan, arad;

  // set default stream parameters to previously defined output file stream
  f1.precision(3);
  f1.setf(ios::scientific,ios::floatfield);
  f1.setf(ios::left,ios::adjustfield);

  // setup constant values
  r_p[0].set(0, p[0].toh + p[0].com, 0);     // position vectors for center of mass
  r_p[1].set(0, 0, p[1].toh + p[1].com);
  r_p[2].set(0, -(p[2].toh + p[2].com), 0);
  r_p[3].set(0, 0, -(p[3].toh + p[3].com));
```

```
for (int i=0; i<4; i++)
  r_p[i] /= 100.0;                        // convert positions to meters

for (i=0; i<4; i++)
  if (p[i].occupied)
    inertia += p[i].I + p[i].mass*
                pow((p[i].com + p[i].toh)/100.0,2.0); // riders inertia
inertia += cfuge.I;                       // add centrifuge inertia to riders inertia

Matrix I_c(inertia, 0, 0,                          // centrifuge plus rider inertia tensor
             0, 0, 0,
             0, 0, 0);

r_cfuge.set(cfuge.com, 0, 0);             // centrifuge center of mass position
Matrix invI_ss = minv(shuttle.I);                  // inverse of shuttle inertia tensor

// write header information to output file

// main simulation loop
for (time = 0; time < 60.0*endTime2; time += dt)
  {
  // save old values for use in numerical integrations
  omega_cLast = omega_c;
  alpha_cLast = alpha_c;
  omega_ssLast = omega_ss;
  alpha_ssLast = alpha_ss;
  v_ssLast = v_ss;
  a_ssLast = a_ss;

  // check to see if centrifuge needs to accelerate to reach appropriate
  // steady state angular velocity to generate maxG at subject #1 center
  // of mass.  Note if subject #1 not occupied, the loop will use default
  // values from subject #1 to accelerate.  The tangential acceleration is
  // limited by the maxTanG variable.
  if (time <= 60.0*endTime1)
   {
   if (mag(omega_c ^ (omega_c ^ r_pr[0]))/g <= maxG)
       alpha_c.set(alpha_c.x() + onsetG*g*dt,0,0);
   else
       alpha_c.set(0,0,0);
   if ((mag(alpha_c ^ r_pr[0])/g >= maxTanG) && (mag(alpha_c) != 0))
         alpha_c.set(maxTanG*g/mag(r_pr[0]),0,0);
   }
  else
   {
   if (mag(omega_c) >= 0.02)
       alpha_c.set(alpha_c.x() - onsetG*g*dt,0,0);
   else
       alpha_c.set(0,0,0);
   if ((mag(alpha_c ^ r_pr[0])/g >= maxTanG) && (mag(alpha_c) != 0))
        alpha_c.set(-maxTanG*g/mag(r_pr[0]),0,0);
  }
omega_c += 0.5*(alpha_c + alpha_cLast)*dt;          // cfuge angular velocity
gamma_c += 0.5*(omega_c + omega_cLast)*dt;          // cfuge angular position
if (gamma_c.x() >= 2.0*pi) gamma_c.set_x(gamma_c.x() - 2.0*pi);

// compute rotated position vectors.  Use transpose of rotation matrix
// because actually need to rotate back to fixed coordinate system.
Matrix Rcfuge = setRMatrix(mag(gamma_c), 'x');
for (i=0; i<4; i++)
  {
  r_pr[i] = trans(Rcfuge)*r_p[i];                // position of subject relative to cfuge com
  r_prtotal[i] = r_pr[i] + r_cfuge;              // position relative to shuttle com
  }

// compute total angular velocity and angular acceleration vectors.
omega_tot = omega_ss + omega_c;
alpha_tot = alpha_ss + alpha_c + (omega_ss ^ omega_c);

// compute velocities and accelerations of the subjects' center of masses
// Note ()'s important because overloaded ^ cross product is usually
// lower priority than + sign.
for (i=0; i<4; i++)
  {
  v_p[i] = omega_tot ^ r_prtotal[i];             // subject velocities
  a_p[i] = (alpha_tot ^ r_prtotal[i]) + (omega_tot ^ v_p[i]); // total acceleration
  atan = alpha_c ^ r_pr[i];                      // tangential acceleration
  arad = omega_c ^ (omega_c ^ r_pr[i]);          // radial acceleration
  }

// compute force required to accelerate the subjects.  Note neglected
// acceleration of subjects due to motion of the shuttle -- negligible
// compared to the motion of the centrifuge.
for (i=0; i<4; i++)
  if (p[i].occupied)
      F_p[i] = p[i].mass*a_p[i];

// compute resultant acceleration of the shuttle center of mass due
// to motion of the subjects.
F_ss = 0.0;                                      // reset to 0 each loop
for (i=0; i<4; i++) F_ss += -1.0*F_p[i];         // equal and opposite reactive force
a_ss = F_ss/shuttle.mass;                        // shuttle acceleration
v_ss += 0.5*(a_ss + a_ssLast)*dt;                // shuttle velocity
r_ss += 0.5*(v_ss + v_ssLast)*dt;                // shuttle displacement

// compute moment applied to shuttle center of mass due to acceleration
// of subjects, compute the angular momentum of the shuttle, compute the
// resultant angular acceleration of the shuttle due to centrifuge moments.

M_cfuge = -1.0*I_c*alpha_c;                      // drive torque
M_imb = r_cfuge ^ F_ss;                          // imbalance moments
```

```
    H_c = I_c*omega_c;                              // angular momentum of cfuge
    M_gyro = omega_ss ^ H_ss;                  // gyroscopic restoring torque
    M_ss = M_cfuge + M_imb - M_gyro;                // total moment applied to shuttle
    alpha_ss = invI_ss * M_ss;                 // shuttle angular acceleration
    omega_ss += 0.5*(alpha_ss + alpha_ssLast)*dt; // shuttle angular velocity
    gamma_ss += 0.5*(omega_ss + omega_ssLast)*dt; // shuttle angular displacement

    // write desired variables to output file

    fl << setw(12) << time;
    if (outputData.checkBox & 1)
      fl << " (" << omega_c.x()*rd << ", " << omega_c.y()*rd << ", " << omega_c.z()*rd << ")
";
    if (outputData.checkBox & 2)
      fl << " (" << alpha_c.x()*rd << ", " << alpha_c.y()*rd << ", " << alpha_c.z()*rd << ") ";
    if (outputData.checkBox & 4)
      fl << " (" << M_cfuge.x() << ", " << M_cfuge.y() << ", " << M_cfuge.z() << ") ";
    if (outputData.checkBox & 8)
      fl << " (" << F_ss.x() << ", " << F_ss.y() << ", " << F_ss.z() << ") ";
    if (outputData.checkBox & 16)
      fl << " (" << M_ss.x() << ", " << M_ss.y() << ", " << M_ss.z() << ") ";
    if (outputData.checkBox & 32)
      fl << " (" << a_ss.x() << ", " << a_ss.y() << ", " << a_ss.z() << ") ";
    if (outputData.checkBox & 64)
      fl << " (" << alpha_ss.x()*rd << ", " << alpha_ss.y()*rd << ", " << alpha_ss.z()*rd << ")
";
    if (outputData.checkBox & 128)
      fl << " (" << r_ss.x() << ", " << r_ss.y() << ", " << r_ss.z() << ") ";
    if (outputData.checkBox & 256)
      fl << " (" << gamma_ss.x()*rd << ", " << gamma_ss.y()*rd << ", " << gamma_ss.z()*rd
<< ") ";

    fl << endl;


    }
  f!.close();
}
```

APPENDIX B3
Cardiovascular Simulation Program

B3.0 **Program Name.** CVMODEL.FOR for MS-DOS

B3.1 **Purpose.** CVMODEL estimates the steady state pressures and flows in the cardiovascular system. The program outputs three files: CVAOUT.TXT, CVVOUT.TXT, and CVPOUT.TXT. These files contain the arterial, venous, and peripheral pressures and flows in each of the 50 vascular compartments simulated by the model. The acceleration applied to each compartment is varied by the user by setting parameters in the program.

B3.2 **General Description of Program Flow.** There are four major modules contained in three files. The main program, CVMODEL calls INITIAL to set the initial conditions for the differential equations which are defined in CVSUBS, and solved by RKF45, a Runge-Kutta integrator. In its present form, the user must code the time, and spatial variation in the acceleration and the segment orientation in subroutine INITIAL and/or CVSUBS and recompile the program for execution. The only data file required is a two line file which sets the integration parameters. The operation of the program is fully documented in comment lines contained within the code.

B3.3 **Output Variables.** The output consists of a header in CVAOUT.TXT which documents the integration parameters and the run title line. This is followed by 41 columns of output. Column 1 is time in seconds, column 2-21 contain pressure in Pascals, and columns 22-41 contain flows in $m^3 \cdot sec^{-1}$. CVVOUT.TXT and CVPOUT.TXT contain only the pressure and flow data, i.e. there is no header data.

```
CVMODEL.FOR - December 17, 1993 (10:12 am)

      PROGRAM CVMODEL
C...
C...  PROGRAM CVMODEL CALLS: (1) SUBROUTINE INITAL TO DEFINE THE ODE
C...  INITAL CONDITIONS, (2) SUBROUTINE RKF45 TO INTEGRATE THE ODES,
C...  AND (3) SUBROUTINE PRINT TO PRINT THE SOLUTION.
C...
C...  THE FOLLOWING CODING IS FOR 500 ODES.  IF MORE ODES ARE TO BE INTE-
C...  GRATED, ALL OF THE 500'S SHOULD BE CHANGED TO THE REQUIRED NUMBER
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
      INTEGER NI, NO, NEQN, NSTOP, NORUN
      COMMON/T/          T,     NSTOP,     NORUN,   PP,  TIM
     1      /Y/   Y(500)
     2      /F/   F(500)
C...
C...  THE NUMBER OF DIFFERENTIAL EQUATIONS IS IN COMMON/N/ FOR USE IN
C...  SUBROUTINE FCN
      COMMON/N/      NEQN  ! TWO EQUATIONS PER VASCULAR SEGMENT
C...
C...  COMMON AREA TO PROVIDE THE INPUT/OUTPUT UNIT NUMBERS TO OTHER
C...  SUBROUTINES
      COMMON/IO/      NI,        NO
C...
```

```
C...  ABSOLUTE DIMENSIONING OF THE ARRAYS REQUIRED BY RKF45
C...
C...    THE USER MUST PROVIDE STORAGE IN HIS CALLING PROGRAM FOR THE ARRAYS
C...    IN THE CALL LIST  -      Y(NEQN) , WORK(3+6*NEQN) , IWORK(5)  ,
C...    DECLARE F IN AN EXTERNAL STATEMENT, SUPPLY SUBROUTINE F(T,Y,YP) AND

      DOUBLE PRECISION YV(500), WORK(3500)
      INTEGER IWORK(5)
C...
C...  EXTERNAL THE DERIVATIVE ROUTINE CALLED BY RKF45
C...
      EXTERNAL FCN
C...
C...  ARRAY FOR THE TITLE (FIRST LINE OF DATA), CHARACTERS  END OF RUNS
      CHARACTER TITLE(20)*4, ENDRUN(3)*4
C...
C...  DEFINE THE CHARACTERS  END OF RUNS
      DATA ENDRUN/'END ','OF R','UNS '/
C...
C...  DEFINE THE INPUT/OUTPUT UNIT NUMBERS
      NI=5
      NO=6
C...
C...  OPEN INPUT AND OUTPUT FILES
      OPEN(NI,FILE='CVDATA.DAT')
      OPEN(NO,FILE='CVOPUT.TXT',BLOCKSIZE=2048)
C...
C...  INITIALIZE THE RUN COUNTER
      NORUN=0
C...
C...  BEGIN A RUN
1     NORUN=NORUN+1
C...
C...  INITIALIZE THE RUN TERMINATION VARIABLE
      NSTOP=0
C...
C...  READ THE FIRST LINE OF DATA
C...
      READ(NI,1000,END=999) (TITLE(I), I = 1, 20)
C...
C...  TEST FOR  END OF RUNS  IN THE DATA
C...
      DO 2 I = 1, 3
      IF(TITLE(I) .NE. ENDRUN(I)) GO TO 3
2     CONTINUE
C...
C...  AN END OF RUNS HAS BEEN READ, SO TERMINATE EXECUTION
999   STOP
C...
C...  READ THE SECOND LINE OF DATA
C...
3     READ(NI,*,END=999) T0, TF, TP
C...
C...  READ THE THIRD LINE OF DATA
C...
      READ(NI,*,END=999) NEQN, ERROR
C...
C...  PRINT A DATA SUMMARY
      WRITE(NO,1003)NORUN,(TITLE(I), I = 1, 20),
     1                    T0, TF, TP,
     2                    NEQN, ERROR
      WRITE(*,1003) NORUN, (TITLE(I), I = 1, 20),
     1                    T0, TF, TP,
     2                    NEQN, ERROR
C...
C...  INITIALIZE TIME
      T = T0
C...
C...  SET THE INITIAL CONDITIONS
```

```
      CALL INITAL
C...
C...  SET THE INITIAL DERIVATIVES (FOR POSSIBLE PRINTING)
      CALL DERV
C...
C...  PRINT THE INITIAL CONDITIONS
      CALL PRINT(NI, NO)
C...
C...  SET THE INITIAL CONDITIONS FOR SUBROUTINE RKF45
      TV = TO
      DO 5 I = 1, NEQN
      YV(I) = Y(I)
5     CONTINUE
C...
C...  SET THE PARAMETERS FOR SUBROUTINE RKF45
C...
C...   FIRST CALL TO RKF45
C...
      RELERR = ERROR
      ABSERR = ERROR
      IFLAG = 1
      TOUT = TO + TP
C...
C...  CALL SUBROUTINE RKF45 TO START THE SOLUTION FROM THE INITIAL
C...  CONDITION (IFLAG = 1) OR COMPUTE THE SOLUTION TO THE NEXT PRINT
C...  POINT (IFLAG = 2)
C...
4     CALL RKF45(FCN,NEQN,YV,TV,TOUT,RELERR,ABSERR,IFLAG,WORK,IWORK)
C...
C...  PRINT THE SOLUTION AT THE NEXT PRINT POINT
C...
      T=TV
      TOUT = TV + TP
      PRINT *,"Time = ", T
      DO 6 I = 1, NEQN
         Y(I) = YV(I)
6     CONTINUE
      CALL DERV
      CALL PRINT(NI,NO)
C...
C...  TEST FOR AN ERROR CONDITION
      IF(IFLAG .NE. 2) THEN
C...
C...     PRINT A MESSAGE INDICATING AN ERROR CONDITION
         WRITE(NO,1004) IFLAG
C...
C...     GO ON TO THE NEXT RUN
         GO TO 1
      END IF
C...
C...  CHECK FOR A RUN TERMINATION
      IF(NSTOP .NE. 0) GO TO 1
C...
C...  CHECK FOR THE END OF THE RUN
C...
      IF(TV .LT. (TF - 0.5D0*TP)) GO TO 4
C...
C...  THE CURRENT RUN IS COMPLETE, SO GO ON TO THE NEXT RUN
      GO TO 1
C...
C...  ******************************************************************
C...
C...  FORMATS
C...
1000  FORMAT(20A4)
1001  FORMAT(3E10.0)
1002  FORMAT(I5,20X,E10.0)
1003  FORMAT(1H1,
     1 ' RUN NO. - ',I3,2X,20A4,//,
     2 ' INITIAL T - ',E10.3,//,
     3 '   FINAL T - ',E10.3,//,
     4 '   PRINT T - ',E10.3,//,
     5 ' NUMBER OF DIFFERENTIAL EQUATIONS - ',I5,//,
     6 ' MAXIMUM INTEGRATION ERROR - ',E10.3,//,
     7 1H1)
1004  FORMAT(1H ,//,' IFLAG = ',I3,//,
     1 ' INDICATING AN INTEGRATION ERROR, SO THE CURRENT RUN'      ,/,
     2 ' IS TERMINATED.  PLEASE REFER TO THE DOCUMENTATION FOR'   ,/,
     3 ' SUBROUTINE',//,25X,'RKF45',//,
     4 ' FOR AN EXPLANATION OF THESE ERROR INDICATORS'             )
      END
      SUBROUTINE FCN(TV,YV,YDOT)
C...
C...  SUBROUTINE FCN IS AN INTERFACE ROUTINE BETWEEN SUBROUTINES RKF45
C...  AND DERV
C...
C...  NOTE THAT THE SIZE OF ARRAYS Y AND F IN THE FOLLOWING COMMON AREA
C...  IS ACTUALLY SET BY THE CORRESPONDING COMMON STATEMENT IN MAIN
C...  PROGRAM HEADHIT
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
      INTEGER NEQN, NSTOP, NORUN
      COMMON/T/          T,     NSTOP,     NORUN
     1      /Y/      Y(500)
     2      /F/      F(500)
C...
C...  THE NUMBER OF DIFFERENTIAL EQUATIONS IS AVAILABLE THROUGH COMMON
C...  /N/
C...
      COMMON/N/      NEQN
C...
C...  ABSOLUTE DIMENSION THE DEPENDENT VARIABLE, DERIVATIVE VECTORS
      DOUBLE PRECISION YV(500), YDOT(500)
C...
C...  TRANSFER THE INDEPENDENT VARIABLE, DEPENDENT VARIABLE VECTOR
C...  FOR USE IN SUBROUTINE DERV
C...
      T = TV
      DO 1 I = 1, NEQN
         Y(I) = YV(I)
1     CONTINUE
C...
C...  EVALUATE THE DERIVATIVE VECTOR
C...
      CALL DERV
C...
C...  TRANSFER THE DERIVATIVE VECTOR FOR USE BY SUBROUTINE RKF45
C...
      DO 2 I = 1, NEQN
         YDOT(I) = F(I)
2     CONTINUE
      RETURN
      END
```

```
SUBROUTINE CVSUBS.FOR - December 17, 1993 (10:30 am)

DECK CVSUBS.FOR - SUBROUTINES REQUIRED TO IMPLEMENT A DYNAMIC MODEL OF THE HUMAN
C...  CARDIOVASCULAR SYSTEM
C...
C...  LAST REVISION:  12/01/93
C...
      SUBROUTINE INITIAL
C...
C...  THE model described herein parallels the development presented in a paper by
C...  White, RJ, Cronton, DG and Fitzgerald, DG.  Cardiovascular Modelling:  Simulating
C...  the Human Cardiovascular Response to Excercise, Lower Body Negative Pressure, Zero
C...  Gravity and Clinical Conditions.  Adv. Cardiovasc. Phys. (Part I), pp. 195-229 (Karger,
C...  Basel 1983).  It also draws from papers by Jaron, et al who took a similar approach
C..   In particular many of the parameter values for the physical properties of the segments
C...  were taken from:
C...
C...       Jaron, D, Moore, TW, and Bai, J.  Cardiovascular Response to Acceleration Stress:
C...       A Computer Simualtion.  Proceedings of the IEEE, Vol 76, No 6, pp. 700-707 (1988).
C...
C...  However, some of the parameters listed in Jaron were clearly in error.  Where new
C...  parameters were requred they were derived to yield generally acceptable
C...  flow/pressure/volume and compliance characteristics of the various
C...  cardiovascular subdivisions.  In particular, data from
C...
C...       Burton, Alan, C.  Physiology and Biophysics of the Circulation.  Year
C...       Book Medical Publishers, Inc.  Chicago, IL (1965)
C...
C...     and,
C...
C...       Guyton, A.C.  Textbook of Medical Physiology.  7th Ed.,
C...       W.B. Saunders, Philadelphia, PA. (1985).
C...
C...  The model describes the spatial and temporal variation in the mean
C...  blood pressure along the z-axis of the body.  The model negelects the
C...  non-linear and convective terms in the Navier-Stokes Equation.  The
C...  model also assumes negligible radial flow.  The flow is assumed laminar
C...  except in the ascending an descending aorta where fluid flow resis-
C...  tance multiplied by 33 to account for turbulent pressure losses.
C...
C...  NOTE:  THE SUBSCRIPT NOTATION INDICATES PARTIAL DERIVATIVE WRT THE
C...         SUBSCRIPT  E.G.  Xt ==> THE FIRST PARTIAL OF X WRT TIME
C...         IN THIS MODEL:
C...
C...                    t = Time                          [sec]
C...                    r = Radius of vascular segment    [m]
C...                    l = Length of vascular segment    [m]
C...                 rho0 = density of blood              [kg/m^3]
C...                  mu0 = viscosity of blood            [N-sec/m^2]
C...
C... MODEL FOR ARTERIAL SEGMENTS
C...
C...  The following set of simultaneous equations are solved for each
C...  arterial vascular segment.
C...
C...    Pt(t) = 1/C*(Qin(t) - Qout(t)) + R2*(Qtin -Qtout)
C...    Qt(t) = 1/L*(Pin(t) - Pout(t) + PGz - R*Q(t))
C...    rt(t) = 1/(2*Pi*r*l)*(Qin(t) - Qout(t))
C...
C...    Where,
C...
C...     P = The pressure in the segment                 [Pa]
C...     Q = The segmental volume flow                   [m^3/sec]
C...     C = The capacitance of the segment              [m^3/Pa]
C...     L = The inertance of the segment                [kg/m^4] or [Pa-sec^2/m^3]
C...     Ra = The viscous flow resistance in the segment [m].
C...     PGz = The hydrostatic pressure difference
C...             across the segment because of gravity   [Pa]
C...
```

```
C...
C...    And, the following approximations for Ra, La, and Ca are taken from
C...    a paper by
C...
C...       Rideout, et al.  Difference-Differential Equations for Fluid
C...       Flow in Distensible Tubes.  IEEE Transactions on Bio-Medical
C...       Engineering.  Vol BME-14, NO. 3, pp 171-177. Jul 1967.
C...
C...       Ra = 81*mu0*l/(8*pi*r^4)
C...       La =  9*rho0*l^2/(4*VOL)
C...       Ca =  3*r*VOL*l/(2*E*h)
C...
C...    Where,
C...
C...       E = Young's modulus for vessel wall              [Pa]
C...       h = Vessel Wall thickness                        [m].
C...
C...    Finally,
C...
C...       PGz = rho0*Gz*g0*l*cos(theta).
C...
C...    Where,
C...
C...       Gz    = The z-axis "G-level" in units of earth's gravity    [unitless]
C...       g0    = The earth's gravitational acceleration              [m/sec^2]
C...       theta = The angle between the segment and the z-axis        [radians].
C...
C...
C... MODEL FOR VENOUS SEGMENTS
C...
C...    The model for venous segments was adapted from
C...
C...       Snyder, et al.  Computer Simulation Studies of
C...       Venous Circulation.  IEEE Trans Bio-Med Engr Vol BME-16,
C...       NO. 4 pp 325-334. Oct 1969.
C...
C...    The unstressed internal volume of a vascular segment in assumed to be
C...
C...        V = Pi*r^2*l                          [m^3].
C...
C...    When the contained volume, v, is greater than V, the transmural
C...    pressure is assumed to be related to the contained volume by,
C...
C...        dPwall = 1/C*v.
C...
C...    Where C is the vascular compliance as defined above.  For v < V,
C...
C...        dPwall = 1/(20*C)*v.
C...
C...    In a collapsed or partially collapsed vein the flow-pressure
C...    relationship based on an (assumed) elliptical cross-section
C...    and is given by
C...
C...        Qt = 1/Lv*(Pin(t) - Pout(t) + PGz - Rv*Q(t))
C...
C...    Where,
C...
C...        Lv =  9*rho0*l^2/(4*v), and
C...
C...             | 81*mu0*pi^2*l^4*r^2/(8*v^3) for v < V
C...        Rv = | or
C...             | 81*mu0*l/(8*pi*r^4)     for v > or = V
C...             | or 81*mu0*l^3/(8*v^2)       ".
C...
C...
C...
C...
```

```
C...   PERIPHERAL CAPILLARY BEDS.
C...
C...     Peripheral capillary resistance and capacitance
C...     are modelled as lumped parameter models ("T" circuits)
C...     following the method of Jaron, et al (cited above).
C...
C...
C...
C...                          Ra         Rv
C...                Pa  o--/\/\/\--|--/\/\/\---o Pv
C...                               |
C...                              ---- Cperipheral
C...                              -|-
C...                               |
C...                              Pexternal (ie P G-suit)
C...                               |
C...                              Pref (Atmospheric)
C...
C...
C...
C...    PULMONARY CIRCULATION
C...
C...        The pulmonary circulation is modelled as a lumped
C...        parameter model again following Jaron's method.  The
C...        model is a "PI" circuit as shown below.
C...
C...
C...        Prtvent    Rpul
C...      Qvr>---->|---/\/\/\-----| Pleft atrium
C... venous        |              |
C... return       ---- Cpul       |
C...              -|-             |
C...               |              |
C...         Intra-Thoracic Pressure
C...
C...
C... CARDIAC CIRCULATION/OUTPUT
C...
C...   The chambers of the heart are modelled as variable capacitances
C...   separated by one-way valves.  The pulmonic and aortic valves are
C...   also modelled as one-way valves.  The general method of modelling
C...   The heart and its circulation follows the method of Snyder et al.
C...   Output flow from the left atrium and left ventricle are modelled
C...   as simple half-wave rectified sinusoids whose volume flows are
C...   estimated from pulmonary venous flow.
C...
C...
C... PRESSURE REFERENCE
C...
C...   The pressure reference for the model is located at the tri-cuspid
C...   valve which presumably tracks intrathoracic pressure.
C...
C...
C... MODEL MECHANICS
C...
C...    For each vessel segment, three coupled non-linear differential
C...    equations must be solved simultaneously.  There are 20 vascular
C...    segments (the pulmonary circuit is segment 1).  There are 10
C...    capillary bed segments modelled as simple resistance and compliance
C...    circuits which are affected directly by extra-vascular pressure.
C...    The pressures and flows in the various segments are coupled by
C...    their spatial connection.  The following table gives the
C...    approximate anatomical location and the corresponding z-axis
C...    coordinate (measured from the tricuspid valve) for each segment.
C...    The z-axis coordinates were based on a 177 cm tall standing man.
C...
```

```
C...                                          Arterial
C...                                          Origin
C... Segment            Anatomical            Z-axis          Peripheral
C... Number             Location              coordinate (cm)    Bed
C...==================================================================
C...   1            Mid-Pulmonary                0               X
C...   2            Ascending Aorta              0
C...   3            Descending Aorta             5
C...   4            Thoracic Aorta/Vena Cava    -8               X
C...   5            Diaphragm/Lower Lung       -15
C...   6            Renal/Hepatic              -22               X
C...   7            Splanchnic                 -32               X
C...   8            Buttocks                   -42               X
C...   9            Femoralis                  -50
C...  10            Mid Thigh                  -65               X
C...  11            Knee/Poplateal             - 80
C...  12            Calf                       -100              X
C...  13            Ankle                      -125
C...  14            Foot                       -132              X
C...  15            Aortic Arch                  6
C...  16            Lower Neck                  15
C...  17            Carotid Sinus               25
C...  18            Ophthalmic                  34               X
C...  19            Mid Brain                   37
C...  20            Cerebral                    42               X
C...
C...
C...
C...  Initial Conditions (t = 0)
C...
C...      The initial conditions for pressure, flow, and volume are
C...      set based on a steady-state solution for the model at 1 Gz
C...      for a supine posture.  For other postures, the initial theta's
C...      for the segments must be changed.
C...
C...   Postural and/or Gz changes during a simulation.
C...
C...      This can be most easily accomplished by adding time
C...      varying profiles for Gz and the theta's in SUBROUTINE DERV
C...      which forms the derivatives for CVMODEL.
C...
C... THE NUMERICAL METHOD OF LINES (W.E. SCHIESSER) IS EMPLOYED TO
C... INTEGRATE THE COUPLED DIFFERENTIAL EQUATIONS (DES).
C...
C... ODE COMMON
C...
C...  /Y/ time variables
C...  /F/ time derivatives of variables
C...  /R/ & /I/ real and integer parameters required to define constants and
C...         define the spatial integration grid.
C...
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      PARAMETER (NEQ = 20, NPSEG = 10)
      INTEGER NSTOP, NORUN, IP
      INTEGER*2 ALIN, ALOUT, VLIN, VLOUT, PVS, PIN, POUT
      DOUBLE PRECISION MM2PA, muO
      COMMON/T/        T,     NSTOP,     NORUN    ! Run Parameters
C...
C...  Arrays for segmental variables Pressure (P), Flow(Q), and radii (r)
C...
     1  /Y/    HP(4),      HQ(4),                ! Heart's Chambers RA=1
     *       AP(NEQ),    AQ(NEQ),    Ar(NEQ),    ! Arterial P, Q, r
     *       VP(NEQ),    VQ(NEQ),    Vr(NEQ),    ! Venous P, Q, r
     *       VOP(3),     VOQ(3),                 ! Venous flows into heart
     *       PP(NEQ),    PQ(NEQ),                ! Peripheral P, Qin, Qout
C...
C... Time derivatives of the segmental variables: Pt, Qt, rt
```

```
C...
     2  /F/   HPt(4),     HQt(4),                        ! Heart's Chambers RA=1
     *      APt(NEQ),  AQt(NEQ),   Art(NEQ),          ! Arterial Pt, Qt, rt
     *      VPt(NEQ),  VQt(NEQ),   Vrt(NEQ),          ! Venous Pt, Qt, rt
     *      VOPt(3),    VOQt(3),                       ! Venous flows into heart
     *      PPt(NEQ),  PQt(NEQ),                       ! Peripheral Pt, QINt, QOUTt
C...
C... Parameters necessary to form the differential equations
C...
C...    Pi -  3.14159...
C...    g0 -  9.80665 [m/sec^2]  earth's acceleration of gravity
C...  rho0 -  1050.   [kg/m^3]   density of whole blood (45% Hct)  ¦ Assumed
C...   mu0 -  2.7     [cp]       viscosity of whole blood (45% Hct)¦ Constant
C...   D2R -  pi/180  [radians/degree] scale factor
C...
     3  /R/  Pi, g0, rho0, mu0,  D2R, MM2PA, R2,       ! Constants
     *       ZAO(NEQ), ZAT(NEQ), ZVO(NEQ), ZVT(NEQ),   ! Arterial & Venous
     *       THETA(NEQ), HCAP(4), HVOL(4), HVR(4),    ! Orientation angle
     *       ALO(NEQ),  ArU(NEQ),  AE(NEQ), Ah(NEQ),   ! Arterial l, r, E, h
     *       ACAP(NEQ), ARES(NEQ), AINERT(NEQ),        ! Arterial Capacitance, resistance
     *       VLO(NEQ),  Vr0(NEQ),  VrU(NEQ),           ! Venous l, r, rUNSTRESSED
     *       VCAPO(NEQ), VCAP(NEQ), VRES(NEQ), VINERT(NEQ), ! Venous Capacitance, resistance, inertance
     *       PRA(NEQ), PRV(NEQ), PCAP(NEQ),            ! Peripheral Ra, Rv, C
     *       PINERT(NEQ), PVOL(NEQ),                   ! Peripheral I, V
     *       PAO(NEQ),  PVO(NEQ),                      ! Initial P conditions
     *       QAO(NEQ),  QVO(NEQ),                      ! Initial Q conditions
     *       AVOL(NEQ),  VVOL(NEQ),                    ! A & V Volumes
     *       PEXT(NEQ),                                ! Externally applied Pressure
     *       TO, GSTART, GMAX, TBRK1, TBRK2, TMAX, GFIN, ! G-Profile parameters
     *       Gz, ADPG(NEQ), VDPG(NEQ)                  ! Gz & Delta P from G
     4  /I/  IP, NDXPER(NPSEG),                        ! Peripheral Bed Indexes
     *       ALIN(NEQ), ALOUT(NEQ),                    ! Linkage Data arterial
     *       VLIN(NEQ), VLOUT(NEQ),                    ! Venous
     *       PIN(NEQ), POUT(NEQ),                      ! Peripheral
     *       PVS(NEQ),                                 ! Number parallel venous segments
     *       IFOOTSEG, IHEADSEG, IHEARTSEG             ! Foot, Head, & Heart seg nums
C...
C...  Note that artery output flow feeds artery input or peripheral bed
C...  input, artery input comes only from arteries or the heart, venous
C...  output flows to veins or the heart, venous input comes from veins
C...  and/or peripheral beds, peripheral beds are fed only by arteries, and
C...  feed only veins.  The index (i) for an segment refers to its input
C...  flow and pressure.  The output pressure for a segment is
C...  stored in P(i+1) and its output flow is stored at Q(i+1)
C...
C...
C...  Define Some Constants and Parameters
C...
C...  Heart valve resistances
C...
      DATA HVR/ 1.48D+6, 1.48D+6, 2.96D+6, 2.96D+6/
C...
C...  Heart chamber capacitances
C...
      DATA HCAP/ 2.25D-7, 6.55D-7, 1.12D-7, 3.28D-7/
C...
C...  Arterial Segment length [meters]
C...
      DATA ALO/ 3.D-2,  3.D-2,  5.D-2,  7.D-2,  5.D-2,
     *         12.D-2, 10.D-2,  8.D-2, 15.D-2, 20.D-2,
     *         15.D-2, 25.D-2,  5.D-2,  2.D-2,  9.D-2,
     *         10.D-2,  9.D-2,  3.D-2,  4.D-2,  1.D-2/
C...
C...  Arterial Segment radii at t = 0  [meters]
C...  These radii are based on flow resistance
```

```
C...
      DATA ArU /6.00D-3, 1.00D-2, 1.00D-2, 1.20D-2, 1.00D-2,
     *          6.00D-3, 5.50D-3, 4.00D-3, 3.50D-3, 4.00D-3,
     *          2.80D-3, 4.50D-3, 2.75D-3, 1.00D-3, 6.00D-3,
     *          3.00D-3, 3.00D-3, 3.00D-3, 3.00D-3, 1.50D-3/
C...
C...  These radii are based on capacitance and resistance
C...
C...  Young's modulus for arterial segment walls [Pa]
C...
      DATA AE /  2.5D+5,  5.0D+5,  5.0D+5,  7.0D+5,  7.0D+5,
     *           8.0D+5,  8.0D+5,  8.0D+5,  8.0D+5,  1.0D+6,
     *           1.0D+6,  1.0D+6,  1.0D+6,  1.0D+6,  8.0D+5,
     *           8.0D+5,  8.0D+5,  8.0D+5,  8.0D+5,  8.0D+5/
C...
C...  Wall thickness for arterial segments [meters]
C...
      DATA Ah /  2.D-4,  16.D-4, 16.D-4,  14.D-4, 12.D-4,
     *          12.D-4,  10.D-4, 10.D-4,   8.D-4,  8.D-4,
     *           8.D-4,   6.D-4,  6.D-4,   5.D-4,  5.D-4,
     *           6.D-4,   6.D-4,  6.D-4,   5.D-4,  5.D-4/
C...
C...  Number of parallel venous paths in each segment
C...
      DATA PVS/  4,  1,  1,  2,  4,  4,  4,  4,  2,  2,
     *           2,  2,  4,  4,  2,  2,  4,  4,  4,  4/
C...
C...  Venous segment length [meters]
C...
      DATA VLO/ 10.D-2,  2.D-2,  4.D-2, 10.D-2, 20.D-2,
     *          48.D-2, 40.D-2, 32.D-2, 30.D-2, 40.D-2,
     *          30.D-2, 40.D-2, 30.D-2,  8.D-2, 18.D-2,
     *          20.D-2, 36.D-2, 12.D-2, 16.D-2,  4.D-2/
C...
C...  Initial radii for Venous segment [meters]
C...
      DATA VrU/ 5.0D-3,  5.0D-3, 5.0D-3,  3.0D-3,  3.0D-3,
     *          3.0D-3,  3.0D-3, 3.25D-3,  3.0D-3,  3.5D-3,
     *          3.5D-3,  3.3D-3, 2.5D-3,  1.8D-3,  4.5D-3,
     *          4.0D-3,  4.0D-3, 3.3D-3,  3.0D-3,  2.0D-3/
C...
C...   Venous capacitance [m^3/Pa]
C...
      DATA VCAPO/ 5.0D-7,  5.00D-8,  1.00D-7,  5.00D-7,  5.0D-7,
     *            5.0D-7,  5.00D-7,  4.00D-7,  3.00D-7,  2.5D-7,
     *            3.0D-7,  5.00D-7,  5.00D-8,  5.00D-9,  2.0D-7,
     *            5.0D-7,  8.00D-7,  1.00D-7,  1.00D-7,  3.0D-8/
C...
C...  Peripheral Segment Indices (There are 10 peripheral segments).
C...
      DATA NDXPER / 1, 3, 6, 7, 8, 10, 12, 14, 18, 20/
C...
C...  Peripheral Vascular Capacitance [m^3/Pa]
C...
      DATA PCAP/ 1.13D-7,  0.0D000,  3.75D-8,  0.0000,  0.0D000,
     *           7.50D-8,  1.13D-7,  7.50D-8,  0.0000,  3.75D-8,
     *           0.0D000,  3.75D-8,  0.0D000,  3.75D-8, 0.0D000,
     *           0.0D000,  0.0D000,  3.75D-8,  0.0D000, 3.75D-8/
C...
C...  Peripheral Vascular Resistance Arterial side [Pa-sec/m^3]
C...  Assign 9.9099 to segments with no peripheral bed.
C...
      DATA PRA/  1.20D+7,  9.9D099,  1.41D+9,  9.9D099,  9.9D099,
     *           1.39D+9,  3.44D+8,  1.37D+9,  9.9D099,  1.34D+9,
     *           9.9D099,  2.18D+9,  9.9D099,  6.34D+9,  9.9D099,
     *           9.9D099,  9.9D099,  1.32D+10, 9.9D099,  6.67D+8/
C...
C...  Peripheral Vascular Resistance Venous side [Pa-sec/m^3]
C...  Assign 9.9D99 to segments with no peripheral bed.
```

```
C...
      DATA PRV/  1.33D+6,  9.99D99, 1.56D+8,  9.99D99,  9.99D99,
     *           1.54D+8,  3.83D+7, 1.52D+8,  9.99D99,  1.49D+8,
     *           9.99D99,  2.43D+8, 9.99D99,  7.04D+8,  9.99D99,
     *           9.99D99,  9.99D99, 1.46D+9,  9.99D99,  7.42D+7/
C...
C...  Initialize extramural pressure vector
C...
      DATA PEXT/ 0.0D+2,  0.0D+2,  0.0D+2,  0.0D+2,  0.0D+2,
     *           0.0D+2,  0.0D+2,  0.0D+2,  0.0D+2,  0.0D+2,
     *           0.0D+2,  0.0D+2,  0.0D+2,  0.0D+2,  0.0D+2,
     *           0.0D+2,  0.0D+2,  0.0D+2,  0.0D+2,  0.0D+2/
C...
C...  Set initial flows [m^3/sec]
C...
C...       Arterial flows
C...
      DATA   AQ/ 8.97D-5,  9.00D-5,  9.00D-5,  6.50D-5,  6.50D-5,
     *           6.50D-5,  5.67D-5,  2.33D-5,  1.50D-5,  1.50D-5,
     *           6.67D-6,  6.67D-6,  1.67D-6,  1.67D-6,  1.67D-6,
     *           1.67D-5,  1.67D-5,  1.67D-5,  1.58D-5,  1.58D-5/
C...
C...      Venous flows
C...
      DATA   VQ/ 9.00D-5,  2.00D-5,  7.50D-5,  7.00D-5,  7.00D-5,
     *           6.50D-5,  6.50D-5,  2.33D-5,  1.00D-5,  1.10D-5,
     *           4.70D-6,  7.50D-6,  4.50D-6,  1.67D-6,  1.67D-5,
     *           1.67D-5,  1.67D-5,  1.67D-5,  1.58D-5,  1.58D-5/
C...
C...     Venous output flows into heart
C...
      DATA  VOQ/ 9.0D-5, 1.7D-5, 7.3D-5/
C...
C...       Peripheral flows
C...
      DATA PQ /   8.97D-5,  0.00D00,  8.33D-6,  0.00D00,  0.00D00,
     *            8.33D-6,  3.33D-5,  8.33D-6,  0.00D00,  8.33D-6,
     *            0.00D00,  5.00D-6,  0.00D00,  1.67D-6,  0.00D00,
     *            0.00D00,  0.00D00,  8.33D-7,  0.00D00,  1.58D-5/
C...
C...  Initial Heart Flows
C...
      DATA HQ /   8.97D-5,  8.97D-5,  8.97D-5,  8.97D-5/
C...
C... The following arrays code the linkage between vascular segments
C... Each segment link element contains the index of the next segment
C... in the cardiovascular tree.  For example, ALIN(3) = 4, which
C... means arterial segment 3 feeds arterial segment 4. Segment -1 codes
C... for terminal peripheral beds for arteries and for the heart for veins.
C...
C...  SEGMENT   A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14
C...
      DATA ALIN/-1, 3, 4, 5, 6, 7, 8, 9,10, 11, 12, 13, 14, -1,
C...
C...  SEGMENT  A15 A16 A17 A18 A19 A20
C...
     *          16, 17, 18, 19, 20, -1/
C...
C... Venous linkage
C...
C...  SEGMENT    V1  V2 V3 V4 V5 V6 V7 V8  V9 V10 V11 V12 V13 V14
C...
      DATA VLIN/-1, -1, -1, 3, 4, 5, 6, 7,  8,  9, 10, 11, 12, 13,
C...
```

```
C...  SEGMENT   V15 V16 V17 V18 V19 V20
C...                                P20
    *            2, 15, 16, 17, 18, 19/
C...
C...  Peripheral Segment Linkage, PIN specifies the input source for the
C...  segment which is only from arteries, POUT specifies the output segment
C...  which is only to veins. "0" means there is no peripheral segment for the
C...  segment number.  Note that the pressure at the inlet of the peripheral
C...  segment is at AP(PIN(i) + 1)
C..
C...  SEGMENT      P1 P2 P3 P4 P5 P6 P7 P8  P9 P10 P11 P12 P13 P14
C...
      DATA PIN  / 0,  0, 3, 0, 0, 6, 7, 8,  0, 10,  0, 12,  0,  0,
C...
C...  SEGMENT   P15 P16 P17 P18 P19 P20
C...
    *            0,  0,  0, 18,  0,  0/
C..
C...  SEGMENT      P1  P2  P3 P4 P5 P6 P7 P8  P9 P10 P11 P12 P13 P14
C...
      DATA  POUT/ 1,  0,   3, 0, 0, 6, 7, 8,  0, 10,  0, 12,  0,  0,
C...
C...  SEGMENT   P15 P16 P17 P18 P19 P20
C...
    *            0,  0,  0, 18,  0,  0/
C...
C...  Z-axis positions relative to tricuspid valve for
C...  the origin of the arterial segment [m]
C...
      DATA ZAO/  0.0D0,   0.0D0,   5.D-2,  -8.D-2,  -15.D-2,
    *          -20.D-2, -32.D-2, -42.D-2, -50.D-2,  -65.D-2,
    *          -85.D-2,-100.D-2,-125.D-2,-130.D-2,    6.D-2,
    *           15.D-2,  25.D-2,  34.D-2,  37.D-2,   41.D-2/
C...
C...  Z-axis positions relative to tricuspid valve for
C...  the termination of the arterial segment [m].
C...
      DATA ZAT/  3.D-2,  1.50-2,   0.0D0, -15.D-2,  -20.D-2,
    *          -32.D-2, -42.D-2, -50.D-2, -65.D-2,  -85.D-2,
    *         -100.D-2,-125.D-2,-130.D-2,-132.D-2,   15.D-2,
    *           25.D-2,  34.D-2,  37.L-2,  41.D-2,   42.D-2/
C...
C...  Z-axis positions relative to tricuspid valve for
C...  the origin of the venous segment [m].
C...
      DATA ZVO/ 3.0D-2,1.50-2, -1.5D-2, -15.D-2, -20.D-2,
    *          -32.D-2, -42.D-2, -50.D-2, -65.D-2, -85.D-2,
    *         -100.D-2,-125.D-2,-130.D-2,-132.D-2,  15.D-2,
    *           25.D-2,  34.D-2,  37.D-2,  41.D-2,  42.D-2/
C...
C...  Z-axis positions relative to tricuspid valve for
C...  the termination of the venous segment [m].
C...
       DATA ZVT/ 0.0D0,  0.0D00,  0.0D00, -1.5D-2,  -15.D-2,
    *          -20.D-2, -32.D-2, -42.D-2, -50.D-2,  -65.D-2,
    *          -85.D-2,-100.D-2,-125.D-2,-130.D-2,   1.5D-2,
    *           15.D-2,  25.D-2,  34.D-2,  37.D-2,   41.D-2/
C...
C... Initial orientations (degrees) of z-axis projection of vascular
C... segments.  Arterial and venous assumed to be at the same orientation.
C...
      DATA THETA/ 0.D0, 0.D0, 0.D0, 0.D0,  0.D0,
    *             0.D0, 0.D0, 0.D0, 0.D0,   0.D0,
    *             0.D0, 0.D0, 0.D0, 0.D0,  0.D0,
    *             0.D0, 0.D0, 0.D0, 0.D0,  0.D0/
C...
C... Define some physical constants
```

```
C...
      Pi     = DACOS(-1.0D0)              ! pi
      g0     = 9.80665D0                  ! m/sec^2 - earth's gravity
      rho0   = 1050.0D0                   ! kg/m^3  - density of whole blood
      mu0    = 2.70D-3                    ! N-sec/m^2 - fluid viscosity
      D2R    = Pi/180.D0                  ! scale from degrees to radians
      MM2PA  = 1.0132505/760.D0           ! scale from mmHg to Pascals
      Gz     = 0.D0                       ! Initial Gz = 0
      R2     = 0.002D0                      ! Jaron's wall energy term
C...
C...  INITIAL CONDITIONS (T = 0)
C...
C...  The pressures are set assuming a prone posture ie transverse g0.
C...  The initial pressures are assigned by linearly interpolating
C...  pressures from the foot to the heart and from the heart to the
C...  cerebral segment. The mean arterial pressure is assumed
C...  to be 100 mmHg at the heart and 95 mm Hg at both the foot and
C...  cerebral segments.  The venous pressure is assumed to be 2 mmHg
C...  at the heart and 5 mmHg at both the foot and cerebral segments.
C...
      PAFOOT    =  95.D0*MM2PA
      PAHEAD    =  95.D0*MM2PA
      IFOOTSEG  = 14
      IHEADSEG  = 20
      IHEARTSEG =  1
      PRATRM =    PEXT(1)         ! Right Atrial Pressure
      HP(1)  = PRATRM + MM2PA     ! Inlet Pressure to Right Atrium
      HDP1   =    2.D0*MM2PA      ! Increase in right atrial pressure
      PRVENT =  HP(1) + HDP1      ! Inlet Pressure to Right Ventricular
      HP(2)  = PRVENT
      HDP2   = 11.D0*MM2PA        ! Increase in right ventricular pressure
      PLATRM =  4.D0*MM2PA       ! Left Atrial Pressure
      HP(3)  = PLATRM             ! Left Atrial Inlet Pressure
      HDP3   = 4.D0*MM2PA         !
      PLVENT = HP(3) + HDP3       ! Left Ventricular Pressure
      HP(4)  = PLVENT             ! Inlet Pressure to LV
      HDP4   = 92.D0*MM2PA        ! Increase in LV Pressure
C...
C...  G-Profile parameters
C...
      T0 = 0.D0
      GSTART = 0.D0
      GMAX = 3.D0
      TMAX = 55.D0
      TBRK1 = 5.D0
      TBRK2 = 40.D0
      GFIN =  3.D0
      i = 1
      DO WHILE (i .LE. NEQ)
         THETA(i) = THETA(i)*D2R
         i = i + 1
      END DO
      i = 1
      DO WHILE (i .LE. NEQ)
         SCALEF = 1.D0
         IF( i .LE. 5 .OR. i .EQ. 15 ) SCALEF = 33.D0
         ARES(i) = 81.D0*mu0*AL0(i)/(8.D0*Pi*ArU(i)**4)*SCALEF
         DPVS = DBLE(PVS(i))
         Vr(i) = DMAX1(Vr(i), 1.33D0*VrU(i))
         VRES(i) = 81.D0*mu0*(VL0(i)/DPVS)/(8.D0*Pi*Vr(i)**4)
         i = i + 1
      END DO
      AP(1) = HP(2) + HDP2 + 2.D0*MM2PA
      AP(2) = HP(4) + HDP4 + 2.D0*MM2PA
      i = 3
      DO WHILE ( i .LE. IFOOTSEG)
         AP(i) = AP(i-1) - AQ(i-1)*ARES(i-1)
     *              + rhoO*Gz*gO*(ZAO(i)-ZAT(i))*DCOS(THETA(i))
         i = i + 1
      END DO
      AP(15) = AP(3)
      i = 16
      DO WHILE ( i .LE. IHEADSEG)
         AP(i) = AP(i-1) - AQ(i-1)*ARES(i-1)
     *              + rhoO*Gz*gO*(ZAO(i)-ZAT(i))*DCOS(THETA(i))
         i = i + 1
      END DO
C...
C...  Set the initial radii and arterial capacitance
C...
C
      i = 1
      DO WHILE (i .LE. NEQ)
         dPwall  = AP(i)
         deltaR  = 7.5D-1*dPwall*ArU(i)**2/(AE(i)*Ah(i))
         Ar(i)   = ArU(i) + deltaR
         ACAP(i) = 3.DO*Pi*Ar(i)**3*ALO(i)/(2.DO*AE(i)*Ah(i))
         deltaR  = dPwall*ACAP(i)/(2.DO*Pi*Ar(i)*ALO(i))
         Ar(i)   = ArU(i) + deltaR
         AVOL(i) = Pi*Ar(i)**2*ALO(i)
         i = i + 1
      END DO
C...
C...  Initial Venous Pressures
C...
      VOP(2) = HP(1)
      VOP(3) = HP(1)
      VP(1) = HP(3)  + VQ(1)*VRES(1)
     *        - rhoO*Gz*gO*(ZVO(1) - ZVT(1))*DCOS(THETA(1))
      VP(2) = VOP(2) + VQ(2)*VRES(2)
     *        - rhoO*Gz*gO*(ZVO(2) - ZVT(2))*DCOS(THETA(2))
      VP(3) = VOP(3) + VQ(3)*VRES(3)
     *        - rhoO*Gz*gO*(ZVO(3) - ZVT(3))*DCOS(THETA(3))
      i = 4
      DO WHILE ( i .LE. IFOOTSEG)
         VP(i) = VP(i-1) + VQ(i)*VRES(i)
     *              - rhoO*Gz*gO*(ZVO(i) - ZVT(i))*DCOS(THETA(i))
         i = i + 1
      END DO
      VP(15) = VP(2) + VQ(15)*VRES(15)
     *                 - rhoO*Gz*gO*(ZVO(15) - ZVT(15))*DCOS(THETA(15))
      i = 16
      DO WHILE ( i .LE. IHEADSEG)
         VP(i) = VP(i-1) + VQ(i)*VRES(i)
     *              - rhoO*Gz*gO*(ZVO(i) - ZVT(i))*DCOS(THETA(i))
         i = i + 1
      END DO
C...
C...  Compute the initial resistances
C...
      i = 1
      DO WHILE (i .LE. NEQ)
         VCAP(i) = VCAPO(i)
         dPwall  = VP(i) - PEXT(i)
         deltaR  = dPwall*VCAP(i)/(2.DO*Pi*VrU(i)*VLO(i))
         Vr(i)   = VrU(i) + deltaR
         deltaR  = dPwall*VCAP(i)/(2.DO*Pi*Vr(i)*VLO(i))
         Vr(i)   = VrU(i) + deltaR
         VVOL(i) = Pi*Vr(i)**2*VLO(i)
         i = i + 1
      END DO

C...
C...   Initial Heart flows
```

```
C...
      HQ(1) = 9.D-5
      HQ(2) = 9.D-5
      HQ(3) = 9.D-5
      HQ(4) = 9.D-5

C...
C...  Set initial Peripheral Pressures
C...
      PP(1) =  PQ(1)*PRA(1) + VQ(1)*PRV(1) + VP(1)
      i = 3
      DO WHILE ( i .LE. NEQ)
         IF( PIN(i) .GT. 0 )  THEN
             PP(i) = AP(i+1)
         END IF
         i = i + 1
      END DO
      PP(14) =  PQ(14)*PRA(14) + VQ(14)*PRV(14) + VP(14)
      PP(20) =  PQ(20)*PRA(20) + VQ(20)*PRV(20) + VP(20)
C...
C...   Call DERV to set initial derivatives -- loop to stabilize derivatives
C...
      i = 1
      DO WHILE (i .LE. 100)
         CALL DERV
         i = i + 1
      END DO
      IP=0
      RETURN
      END
C...
      SUBROUTINE DERV
C...
C...  DERV CALCULATES THE TIME DERIVATIVES TO BE INTEGRATED BY RKF45
C...
C...  ODE COMMON
C...
C...   /Y/ time variables
C...   /F/ time derivatives of variables
C...   /S/ spatial derivatives of variables
C...   /R/ & /I/ real and integer parameters required to define constants and
C...            define the spatial integration grid.
C...
       IMPLICIT DOUBLE PRECISION (A-H, O-Z)
       PARAMETER (NEQ = 20, NPSEG = 10)
       INTEGER NSTOP, NORUN, IP
       INTEGER*2 ALIN, ALOUT, VLIN, VLOUT, PVS, PIN, POUT
       DOUBLE PRECISION MM2PA, mu0
       COMMON/T/          T,      NSTOP,      NORUN    ! Run Parameters
C...
C...   Arrays for segmental variables Pressure (P), Flow(Q), and radii (r)
C...
     1  /Y/    HP(4),      HQ(4),                     ! Heart's Chambers RA=1
     *        AP(NEQ),   AQ(NEQ),   Ar(NEQ),        ! Arterial P, Q, r
     *        VP(NEQ),   VQ(NEQ),   Vr(NEQ),        ! Venous P, Q, r
     *        VOP(3),     VOQ(3),                     ! Venous flows into heart
     *        PP(NEQ),   PQ(NEQ),                     ! Peripheral P, Qin, Qout
C...
C...  Time derivatives of the segmental variables: Pt, Qt, rt
C...
     2  /F/   HPt(4),     HQt(4),                         ! Heart's Chambers RA=1
     *       APt(NEQ),  AQt(NEQ),   Art(NEQ),           ! Arterial Pt, Qt, rt
     *       VPt(NEQ),  VQt(NEQ),   Vrt(NEQ),           ! Venous Pt, Qt, rt
     *       VOPt(3),    VOQt(3),                       ! Venous flows into heart
     *       PPt(NEQ),  PQt(NEQ),                         ! Peripheral Pt, QINt, QOUTt
C...
C... Parameters necessary to form the differential equations
C...
C...    Pi -  3.14159...
```

```
C...    g0 -  9.80665 [m/sec^2]  earth's acceleration of gravity
C...  rho0 -  1050.   [kg/m^3]   density of whole blood (45% Hct)  | Assumed
C...   mu0 -  2.7     [cp]       viscosity of whole blood (45% Hct)| Constant
C...   D2R -  pi/180  [radians/degree] scale factor
C...
     3  /R/  Pi, g0, rho0, mu0,  D2R, MM2PA, R2,        ! Constants
     *       ZAO(NEQ), ZAT(NEQ), ZVO(NEQ), ZVT(NEQ),    ! Arterial & Venous
     *       THETA(NEQ), HCAP(4), HVOL(4), HVR(4),                              ! Orientation angle
     *       ALO(NEQ), ArU(NEQ),  AE(NEQ), Ah(NEQ),     ! Arterial l, r, E, h
     *       ACAP(NEQ), ARES(NEQ), AINERT(NEQ),         ! Arterial Capacitance, resistance
     *       VLO(NEQ),  VrO(NEQ),  VrU(NEQ),            ! Venous l, r, rUNSTRESSED
     *       VCAPO(NEQ), VCAP(NEQ), VRES(NEQ), VINERT(NEQ), ! Venous Capacitance, resistance, inertance
     *       PRA(NEQ), PRV(NEQ), PCAP(NEQ),             ! Peripheral Ra, Rv, C
     *       PINERT(NEQ), PVOL(NEQ),                    ! Peripheral I, V
     *       PAO(NEQ),  PVO(NEQ),                       ! Initial P conditions
     *       QAO(NEQ),  QVO(NEQ),                       ! Initial Q conditions
     *       AVOL(NEQ),  VVOL(NEQ),                     ! A & V Volumes
     *       PEXT(NEQ),                                 ! Externally applied Pressure
     *       TO, GSTART, GMAX, TBRK1, TBRK2, TMAX, GFIN, ! G-Profile parameters
     *       Gz, ADPG(NEQ), VDPG(NEQ)                   ! Gz & Delta P from G

     4  /I/  IP, NDXPER(NPSEG),                         ! Peripheral Bed Indexes
     *       ALIN(NEQ), ALOUT(NEQ),                     ! Linkage Data arterial
     *       VLIN(NEQ), VLOUT(NEQ),                     ! Venous
     *       PIN(NEQ), POUT(NEQ),                       ! Peripheral
     *       PVS(NEQ),                                  ! Number parallel venous segments
     *       IFOOTSEG, IHEADSEG, IHEARTSEG              ! Foot, Head, & Heart seg nums
      DIMENSION PQOUT(NEQ)
C...
C...   Right Heart
C...
      PRATRM =   PEXT(1)        ! Right Atrial Pressure
      HP(1)  = PRATRM + MM2PA   ! Inlet Pressure to Right Atrium
      HDP1   =   1.D0*MM2PA     ! Increase in right atrial pressure
      PRVENT =  HP(1) + HDP1    ! Inlet Pressure to Right Ventricular
      HP(2)  = PRVENT
      HDP2   = 11.D0*MM2PA      ! Increase in right ventricular pressure
      PINTHO =  -1.0D0 *MM2PA   ! Intra-thoracic Pressure                        ! Extramural
pressure for the pulmonary bed
C...
C...  Left Heart
C...
      PLATRM =   4.D0*MM2PA     ! Left Atrial Pressure
      HP(3)  = PLATRM           ! Left Atrial Inlet Pressure
      HDP3   = 4.D0*MM2PA       !
      PLVENT = HP(3) + HDP3     ! Left Ventricular Pressure
      HP(4)  = PLVENT           ! Inlet Pressure to LV
      HDP4   = 92.D0*MM2PA      ! Increase in LV Pressure
C...
C...   Ensure pressures are not less than external pressures
C...
      i = 1
      DO WHILE ( i .LE. NEQ )
         IF (VQ(i) .LT. 0.D0) THEN
             VQ(i)  = 0.D0    ! Venous valves
             VQt(i) = 0.D0    ! ==> reverse flow NOT ALLOWED
         END IF
         i = i + 1
      END DO
      i = 1
      DO WHILE (i .LE. 3)
         VOQ(i) = DMAX1(VOQ(i), 0.D0)
         i = i + 1
      END DO
      AQ(1) = DMAX1(AQ(1), 0.D0) ! No reverse flow into the heart
      AQ(2) = DMAX1(AQ(2), 0.D0) ! from arteries or out of the heart
      HQ(1) = DMAX1(HQ(1), 0.D0) ! into veins
      HQ(3) = DMAX1(HQ(3), 0.D0)
C...
```

```
C...  Set the Heart level G level
C...
      CALL TRAPG(T0,GSTART,GMAX,TMAX,TBRK1,TBRK2,GFIN,T,Gz)
C...
C...  Set peripheral inlet pressures to the appropriate
C...  arterial outlet pressures
C...
      i = 3
      DO WHILE ( i .LE. NEQ)
         IF( PIN(i) .GT. 0 )  THEN
             PP(i) = AP(i+1)
             PPt(i) = APt(i+1)
         END IF
         i = i + 1
      END DO
C...
C...  Compute the resistance, capacitance and inertance for the arterial segments
C...
      AP(15) = AP(3)
      i = 1
      DO WHILE (i .LE. NEQ)
         IF ( AP(i) .LE. PEXT(i) ) THEN
            Ar(i) = ArU(i)    ! If arterial pressure is below external
         END IF      ! pressure set radius to minimum (unstressed).
C...
C...   rationalize pressure, radius, and capacitance.
C...
         dPwall = DMAX1(AP(i) - PEXT(i), 0.D0)
         ACAP(i) = 3.D0*Pi*Ar(i)**3*ALO(i)/(2.D0*AE(i)*Ah(i))
         deltaR  = dPwall*ACAP(i)/(2.D0*Pi*Ar(i)*ALO(i))
C         Ar(i)   = ArU(i) + deltaR
         AVOL(i) = Pi*Ar(i)**2*ALO(i)
         SCALEF = 1.D0
         IF( i .LE. 5 .OR. i .EQ. 15 ) SCALEF = 33.D0
         ARES(i)   = 81.D0*mu0*ALO(i)/(8.D0*Pi*Ar(i)**4)*SCALEF
         AINERT(i) =  9.D0*rho0*ALO(i)**2/(4.D0*AVOL(i))
         ADPG(i)   = rho0*Gz*g0*(ZAO(i) - ZAT(i))*DCOS(THETA(i))
         i = i + 1
      END DO
C...
C...  Venous resistance, capacitance and inertance for the venous segments
C...
      i = 1
      DO WHILE (i .LE. NEQ)
         dPwall = DMAX1(VP(i) - PEXT(i),0.D0)
         Vrmax = 1.414214*VrU(i)
         Vr(i)  = DMAX1(Vr(i), VrU(i))
         Vr(i)  = DMIN1(Vr(i), Vrmax)
         DPVS = DBLE(PVS(i))
         VVOL(i) = Pi*Vr(i)**2*VLO(i)
         VRES(i) = 81.D0*mu0*(VLO(i)/DPVS)/(8.D0*Pi*Vr(i)**4)
         VINERT(i) = 9.D0*rho0*VLO(i)**2/(4.D0*VVOL(i))
         VDPG(i) = rho0*Gz*g0*(ZVO(i) - ZVT(i))*DCOS(THETA(i))
         i = i + 1
      END DO
      CAPL = 1.D-2 ! Capillary Length
      i = 1
      DO WHILE( i .LE. NPSEG)
         j = NDXPER(i)
         dPwall = DMAX1(PP(j) - PEXT(j), MM2PA)  ! Min pressure 1 mmHg
         PVOL(j) = PCAP(j) * dPwall
         i = i + 1
      END DO
C...
C...   Define the differential equations describing pressure, flow, and
C...   radius
C...
C...   Pt(t) = 1/C*(Qin(t) - Qout(t))  + R2/C*(Qtin(t)- Qtout(t))
C...   Qt(t) = 1/L*(Pin(t) - Pout(t) + PGz - Pext - R*Q(t))
```

```
C...    rt(t) = 1/(2*Pi*r*l)*(Qin(t) - Qout(t))
C...
C...
C...      Infow to the right atrium HQ(1) = VOQ(2) + VOQ(3)
C...
        IF (VQ(2) .LE. 0.D0) VP(2) = HP(1)
        VOQt(2) = 1.D0/VINERT(2)*(VP(2) - HP(1) + VDPG(2)   ! Outflow from the Superior V.C.
     *            - VRES(2)*VOQ(2))
        IF (VQ(3) .LE. 0.D0) VP(3) = HP(1)
        VOQt(3) = 1.D0/VINERT(3)*(VP(3) - HP(1) + VDPG(3)   ! Outflow from the Inferior V.C.
     *            - VRES(3)*VOQ(3))
        HQt(1) = VOQt(2) + VOQt(3)
        HQ(1) = VOQ(2) + VOQ(3)

C...
C...   Right Atrium   [Heart Segment 1]
C...
      HQ2 = HQ(1) - HQ(2)
      HQ2t = HQt(1) - HQt(2)
      HR2 = 2.D-5/HCAP(1)
      HPt(1) = HQ2t*HR2 + HQ2/HCAP(1)
      HPt(1) = 0.D0
      HPt(2) = 0.D0
      HQ(2) = (HP(1) + HDP1 + MM2PA - HP(2))/HVR(1)
      HQt(2) = (HPt(1) - HPt(2))/HVR(1)
      HQt(2) = HQt(1)
      HQ(2) =  HQ(1)
C...
C...  Right Ventricle   [Heart Segment 2]
C...
      HQ2 = HQ(2) - AQ(1)
      HQ2t = HQt(2) - AQt(1)
      HR2 = 2.D-5/HCAP(2)
      HPt(2) = HQ2t*HR2 + HQ2/HCAP(2)
      HPt(2) = 0.D0
C...
C...   Pulmonary Circulation
C...
C...  Pulmonary Artery   [Arterial Segment 1]
C...
      AQ2  = AQ(1) - PQ(1)
      AQ2t = AQt(1) - PQt(1)
      AR2 = R2/ACAP(1)
      APt(1) = AQ2t*AR2 + AQ2/ACAP(1)
      APt(1) = 0.D0
      Art(1) =  1.D0/(2.D0*Pi*Ar(1)*ALO(1))*AQ2
      Art(1) =  0.D0
      AQ(1) = (HP(2) + HDP2 + 2.D0*MM2PA - AP(1))/HVR(2)
      AQt(1) = (HPt(2) - APt(1))/HVR(2)
      AQ(1) = HQ(2)
C...
C...  Pulmonary Capillary Bed   [Peripheral Segment 1]
C...
      PQt(1) = 1.D0/AINERT(1)*(AP(1) - PP(1) + ADPG(1)
     *           - ARES(1)*PQ(1))
      PQ2 = PQ(1) - VQ(1)
      PPt(1) = PRA(1)*PQt(1) + PQ2/PCAP(1)
C...
C...  Pulmonary Veins
C...
      VQt(1) = (PPt(1) - VPt(1) - PQt(1)*PRA(1))/PRV(1)
      VQ2  = VQ(1)  - HQ(3)
      VQ2t = VQt(1) - HQt(3)
      VR2 = R2/VCAP(1)
      VPt(1) = VQ2t*VR2 + VQ2/VCAP(1)
      Vrt(1) =  1.D0/(2.D0*Pi*Vr(1)*VLO(1))*VQ2
C      Vrt(1) =  0.D0
C...
C...  Left Atrium   [Heart Segment 3]
```

```
C...
      HQt(3) = 1.D0/VINERT(1)*(VP(1) - HP(3) + VDPG(1)
     *         - HQ(3)*VRES(1))
      VOQt(1) = HQt(3)
      HQ2 = HQ(3) - HQ(4)
      HQ2t = HQt(3) - HQt(4)
      HR2 = 2.D-5/HCAP(3)
      HPt(3) = HQ2t*HR2 + HQ2/HCAP(3)
      HPt(3) = 0.D0
C...
C...  Left Ventricle   [Heart Segment 4]
C...
      HQ(4) = (HP(3) + HDP3 + 2.D0*MM2PA - HP(4))/HVR(3)
      HQ(4) = HQ(3)
      HQt(4) = (HPt(3) - HPt(4))/HVR(3)
      HQt(4) = HQt(3)
      HQ2 = HQ(4) - AQ(2)
      HQ2t = HQt(4) - AQt(2)
      HR2 = 2.D-5/ACAP(4)
      HPt(4) = HQ2t*HR2 + HQ2/HCAP(4)
      HPt(4) = 0.D0
C...
C...  Ascending Aortic artery [Arterial Segment 2]
C...
      AQ(2) = (HP(4) + HDP4 + 2.D0*MM2PA - AP(2))/HVR(4)
      AQt(2) = (HPt(4) - APt(2))/HVR(4)
      AQ2 = AQ(2) - AQ(3) - AQ(15)
      AQ2t = AQt(2) - AQt(3) - AQt(15)
      AR2 = R2/ACAP(2)
      APt(2) = AQ2t*AR2 + AQ2/ACAP(2)
      APt(2) = 0.D0
      Art(2) = 1.D0/(2.D0*Pi*Ar(2)*ALO(2))*AQ2
      Art(2) = 0.D0
C...
C...  Descending Aorta   [Arterial Segment 3]
C...
      AQt(3) =  1.D0/AINERT(2)*(AP(2) - AP(3)
     *          +  ADPG(2) - ARES(2)*(AQ(3) + AQ(15))) - AQt(15)
      AQ2  =  AQ(3) - AQ(4)  - PQ(3)
      AQ2t = AQt(3) - AQt(4) - PQt(3)
      AR2 = R2/ACAP(3)
      APt(3) = AQ2t*AR2 + AQ2/ACAP(3)
      AQt(4) = 1.D0/AINERT(3)*(AP(3) - AP(4) + ADPG(3)
     *         - ARES(3)*(AQ(4) + PQ(3))) - PQt(3)
      Art(3) =  1.D0/(2.D0*Pi*Ar(3)*ALO(3))*AQ2
      Art(3) = 0.D0
C...
C...  Form the derivatives for the other arterial segments below the heart.
C...
C...  Thoracic and Cardiac   [Arterial Segment 4]
C...
      AQ2 = AQ(4) - AQ(5)
      AQ2t = AQt(4) - AQt(5)
      AR2 = R2/ACAP(4)
      APt(4) =  1.D0/ACAP(4)*AQ2 + AR2*AQ2t
      AQt(5) =  1.D0/AINERT(4)*(AP(4) - AP(5)
     *       +  ADPG(4) - ARES(4)*AQ(5))
      Art(4) =  1.D0/(2.D0*Pi*Ar(4)*ALO(4))*AQ2
      Art(4) = 0.D0
C...
C...  Diaphragm  [Arterial Segment 5]
C...
      AQ2 = AQ(5) - AQ(6)
      AQ2t = AQt(5) - AQt(6)
      AR2 = R2/ACAP(5)
      APt(5) =  AR2*AQ2t + AQ2/ACAP(5)
      AQt(6) =  1.D0/AINERT(5)*(AP(5) - AP(6)
     *       +  ADPG(5) - ARES(5)*AQ(6))
      Art(5) =  1.D0/(2.D0*Pi*Ar(5)*ALO(5))*AQ2
      Art(5) = 0.D0
C...
C...  Renal - Hepatic   [Arterial Segment 6]
C...
      AQ2 = AQ(6) - AQ(7) - PQ(6)
      AQ2t = AQt(6) - AQt(7) - PQt(6)
      AR2 = R2/ACAP(6)
      APt(6) =  AR2*AQ2t + AQ2/ACAP(6)
      AQt(7) =  1.D0/AINERT(6)*(AP(6) - AP(7)
     *        + ADPG(6) - ARES(6)*(AQ(7) + PQ(6))) - PQt(6)
      Art(6) =  1.D0/(2.D0*Pi*Ar(6)*ALO(6))*AQ2
      Art(6) = 0.D0
C...
C...  Splanchnic  [Arterial Segment 7]
C...
      AQ2 = AQ(7) - AQ(8) - PQ(7)
      AQ2t = AQt(7) - AQt(8) - PQt(7)
      AR2 = R2/ACAP(7)
      APt(7) =  AR2*AQ2t + AQ2/ACAP(7)
      AQt(8) =  1.D0/AINERT(7)*(AP(7) - AP(8)
     *        + ADPG(7) - ARES(7)*(AQ(8) + PQ(7))) - PQt(7)
      Art(7) =  1.D0/(2.D0*Pi*Ar(7)*ALO(7))*AQ2
      Art(7) = 0.D0
C...
C...  Buttocks   [Arterial Segment 8]
C...
      AQ2 = AQ(8) - AQ(9) - PQ(8)
      AQ2t = AQt(8) - AQt(9) - PQt(8)
      AR2 = R2/ACAP(8)
      APt(8) =  1.D0/ACAP(8)*AQ2 + AR2*AQ2t
      AQt(9) =  1.D0/AINERT(8)*(AP(8) - AP(9)
     *        + ADPG(8) - ARES(8)*(AQ(9) + PQ(8))) - PQt(8)
      Art(8) =  1.D0/(2.D0*Pi*Ar(8)*ALO(8))*AQ2
      Art(8) = 0.D0
C...
C...  Femoralis   [Arterial Segment 9]
C...
      AQ2 = AQ(9) - AQ(10)
      AQ2t = AQt(9) - AQt(10)
      AR2 = R2/ACAP(9)
      APt(9) =  AR2*AQ2t + AQ2/ACAP(9)
      AQt(10) =  1.D0/AINERT(9)*(AP(9) - AP(10)
     *        + ADPG(9) - ARES(9)*AQ(10))
      Art(9) =  1.D0/(2.D0*Pi*Ar(9)*ALO(9))*AQ2
      Art(9) = 0.D0
C...
C...  Thigh   [Arterial Segment 10]
C...
      AQ2 = AQ(10) - AQ(11) - PQ(10)
      AQ2t = AQt(10) - AQt(11) - PQt(10)
      AR2 = R2/ACAP(10)
      APt(10) =  AR2*AQ2t + AQ2/ACAP(10)
      AQt(11) =  1.D0/AINERT(10)*(AP(10) - AP(11)
     *        + ADPG(10) - ARES(10)*(AQ(11)+PQ(10))) - PQt(10)
      Art(10) =  1.D0/(2.D0*Pi*Ar(10)*ALO(10))*AQ2
      Art(10) = 0.D0
C...
C...  Knee   [Arterial Segment 11]
C...
      AQ2 = AQ(11) - AQ(12)
      AQ2t = AQt(11) - AQt(12)
      AR2 = R2/ACAP(11)
      APt(11) =  AQ2/ACAP(11) +  AR2*AQ2t
      AQt(12) =  1.D0/AINERT(11)*(AP(11) - AP(12)
     *             + ADPG(11) - ARES(11)*AQ(12))
      Art(11) =  1.D0/(2.D0*Pi*Ar(11)*ALO(11))*AQ2
      Art(11) =  0.D0
C...
C...  Calf   [Arterial Segment 12]
```

```
C...
      AQ2 = AQ(12) - AQ(13) - PQ(12)
      AQ2t = AQt(12) - AQt(13) - PQt(12)
      AR2 = R2/ACAP(12)
      APt(12) =  AR2*AQ2t + AQ2/ACAP(12)
      AQt(13) =  1.D0/AINERT(12)*(AP(12) - AP(13)
     *        + ADPG(12) - ARES(12)*(AQ(13)+PQ(12))) - PQt(12)
      Art(12) =  1.D0/(2.D0*Pi*Ar(12)*ALO(12))*AQ2
      Art(12) = 0.D0
C...
C...  Ankle   [Arterial Segment 13]
C...
      AQ2 = AQ(13) - AQ(14)
      AQ2t = AQt(13) - AQt(14)
      AR2 = R2/ACAP(13)
      APt(13) =  AQ2t*AR2 + AQ2/ACAP(13)
      AQt(14) =  1.D0/AINERT(13)*(AP(13) - AP(14)
     *        + ADPG(13) - ARES(13)*AQ(14))
      Art(13) =  1.D0/(2.D0*Pi*Ar(13)*ALO(13))*AQ2
      Art(13) = 0.D0
C...
C...   Foot   [Arterial Segment 14]
C...
      AQ2 = AQ(14) - PQ(14)
      AQ2t = AQt(14) - PQt(14)
      AR2 = R2/ACAP(14)
      APt(14) = AR2*AQ2t + AQ2/ACAP(14)
      PQt(14) = 1.D0/AINERT(14)*(AP(14) - PP(14) + ADPG(14)
     *          - ARES(14)*PQ(14))
      Art(14) =  1.D0/(2.D0*Pi*Ar(14)*ALO(14))*AQ2
      Art(14) = 0.D0
C...
C...   Peripheral Bed in Foot   [Peripheral Segment 14]
C...
      VQ(14) = DMAX1((PP(14) - VP(14) - PQ(14)*PRA(14))/PRV(14),0.D0)
      PQ2 = PQ(14) - VQ(14)
      PPt(14) = PQt(14)*PRA(14) + PQ2/PCAP(14)
C...
C...  Venous Drainage from Foot   [Venous Segment 14]
C...
      VQt(14) = (F. t(14) - VPt(14) - PQt(14)*PRA(14))/PRV(14)
      VQ2 = VQ(14) - VQ(13)
      VQ2t = VQt(14) - VQt(13)
      VR2 = R2/VCAP(14)
      VPt(14) = VQ2t*VR2 + VQ2/VCAP(14)
      VQt(13) = 1.D0/VINERT(14)*(VP(14) - VP(13) + VDPG(14)
     *          - VRES(14)*VQ(13))
      Vrt(14) =  1.D0/(2.D0*Pi*Vr(14)*VLO(14))*VQ2
C...
C...  Ankle  [Venous Segment 13]
C...
      PQOUT(12) = (PP(12) - VP(12) - PRA(12)*PQ(12))/PRV(12)
      PQOUTt = (PPt(12) - VPt(12) - PRA(12)*PQt(12))/PRV(12)
      VQ2 =  VQ(13) - (VQ(12) - PQOUT(12))
      VQ2t = VQt(13) - (VQt(12) - PQOUTt)
      VR2  = R2/VCAP(13)
      VPt(13) = VQ2t*VR2 + VQ2/VCAP(13)
      VQt(12) = 1.D0/VINERT(13)*(VP(13) - VP(12) + VDPG(13)
     *          - VRES(13)*(VQ(12) - PQOUT(12))) + PQOUTt
      Vrt(13) =  1.D0/(2.D0*Pi*Vr(13)*VLO(13))*VQ2
C...
C...  Calf   [Venous Segment 12]
C...
      VQ2 = VQ(12) - VQ(11)
      VQ2t = VQt(12) - VQt(11)
      VR2 = R2/VCAP(12)
      VPt(12) = VQ2t*VR2 + VQ2/VCAP(12)
      VQt(11) = 1.D0/VINERT(12)*(VP(12) - VP(11) + VDPG(12)
     *          - VRES(12)*VQ(11))
```

```
      Vrt(12) =  1.D0/(2.D0*Pi*Vr(12)*VL0(12))*VQ2
C...
C...  Knee   [Venous Segment 11]
C...
      PQOUT(10) = (PP(10) - VP(10) - PRA(10)*PQ(10))/PRV(10)
      PQOUTt = (PPt(10) - VPt(10) - PRA(10)*PQt(10))/PRV(10)
      VQ2 =  VQ(11) - (VQ(10) - PQOUT(10))
      VQ2t = VQt(11) - (VQt(10) - PQOUTt)
      VR2  = R2/VCAP(11)
      VPt(11) = VQ2t*VR2 + VQ2/VCAP(11)
      VQt(10) = 1.D0/VINERT(11)*(VP(11) - VP(10) + VDPG(11)
     *          - VRES(11)*(VQ(10) - PQOUT(10))) + PQOUTt
      Vrt(11) =  1.D0/(2.D0*Pi*Vr(11)*VL0(11))*VQ2
C...
C...  Thigh   [Venous Segment 10]
C...
      VQ2 = VQ(10) - VQ(9)
      VQ2t = VQt(10) - VQt(9)
      VR2 = R2/VCAP(10)
      VPt(10) = VQ2t*VR2 + VQ2/VCAP(10)
      VQt(9) = 1.D0/VINERT(10)*(VP(10) - VP(9) + VDPG(10)
     *          - VRES(10)*VQ(9))
      Vrt(10) =  1.D0/(2.D0*Pi*Vr(10)*VL0(10))*VQ2
C...
C...   Femoralis  [Venous Segment 9]
C...
      PQOUT(8) = (PP(8) - VP(8) - PRA(8)*PQ(8))/PRV(8)
      PQOUTt = (PPt(8) - VPt(8) - PRA(8)*PQt(8))/PRV(8)
      VQ2 =  VQ(9) - (VQ(8) - PQOUT(8))
      VQ2t = VQt(9) - (VQt(8) - PQOUTt)
      VR2  = R2/VCAP(9)
      VPt(9) = VQ2t*VR2 + VQ2/VCAP(9)
      VQt(8) = 1.D0/VINERT(9)*(VP(9) - VP(8) + VDPG(9)
     *          - VRES(9)*(VQ(8) - PQOUT(8))) + PQOUTt
      Vrt(9) =  1.D0/(2.D0*Pi*Vr(9)*VL0(9))*VQ2
C...
C...   Buttocks   [Venous Segment 8]
C...
      PQOUT(7) = (PP(7) - VP(7) - PRA(7)*PQ(7))/PRV(7)
      PQOUTt = (PPt(7) - VPt(7) - PRA(7)*PQt(7))/PRV(7)
      VQ2 =  VQ(8) - (VQ(7) - PQOUT(7))
      VQ2t = VQt(8) - (VQt(7) - PQOUTt)
      VR2  = R2/VCAP(8)
      VPt(8) = VQ2t*VR2 + VQ2/VCAP(8)
      VQt(7) = 1.D0/VINERT(8)*(VP(8) - VP(7) + VDPG(8)
     *          - VRES(8)*(VQ(7) - PQOUT(7))) + PQOUTt
      Vrt(8) =  1.D0/(2.D0*Pi*Vr(8)*VL0(8))*VQ2
C...
C...   Splanchnic   [Venous Segment 7]
C...
      PQOUT(6) = (PP(6) - VP(6) - PRA(6)*PQ(6))/PRV(6)
      PQOUTt = (PPt(6) - VPt(6) - PRA(6)*PQt(6))/PRV(6)
      VQ2 =  VQ(7) - (VQ(6) - PQOUT(6))
      VQ2t = VQt(7) - (VQt(6) - PQOUTt)
      VR2  = R2/VCAP(7)
      VPt(7) = VQ2t*VR2 + VQ2/VCAP(7)
      VQt(6) = 1.D0/VINERT(7)*(VP(7) - VP(6) + VDPG(7)
     *          - VRES(7)*(VQ(6) - PQOUT(6))) + PQOUTt
      Vrt(7) =  1.D0/(2.D0*Pi*Vr(7)*VL0(7))*VQ2
C...
C...   Renal - Hepatic   [Venous Segment 6]
C...
      VQ2 = VQ(6) - VQ(5)
      VQ2t = VQt(6) - VQt(5)
      VR2 = R2/VCAP(6)
      VPt(6) = VQ2t*VR2 + VQ2/VCAP(6)
      VQt(5) = 1.D0/VINERT(6)*(VP(6) - VP(5) + VDPG(6)
     *          - VRES(6)*VQ(5))
      Vrt(6) =  1.D0/(2.D0*Pi*Vr(6)*VL0(6))*VQ2
```

```
C...
C...   Diaphragm   [Venous Segment 5]
C...
      VQ2 = VQ(5) - VQ(4)
      VQ2t = VQt(5) - VQt(4)
      VR2 = R2/VCAP(5)
      VPt(5) = VQ2t*VR2 + VQ2/VCAP(5)
      VQt(4) = 1.D0/VINERT(5)*(VP(5) - VP(4) + VDPG(5)
     *             - VRES(5)*VQ(4))
      Vrt(5) =  1.D0/(2.D0*Pi*Vr(5)*VLO(5))*VQ2
C...
C...   Thoracic Circulation [Venous Segment 4]
C...
        PQOUT(3) = (PP(3) - VP(3) - PRA(3)*PQ(3))/PRV(3)
        PQOUTt = (PPt(3) - VPt(3) - PRA(3)*PQt(3))/PRV(3)
        VQ2 =  VQ(4) - (VQ(3) - PQOUT(3))
        VQ2t = VQt(4) - (VQt(3) - PQOUTt)
        VR2  = R2/VCAP(4)
        VPt(4) = VQ2t*VR2 + VQ2/VCAP(4)
        VQt(3) = 1.D0/VINERT(4)*(VP(4) - VP(3) + VDPG(4)
     *             - VRES(4)*(VQ(3) - PQOUT(3))) + PQOUTt
        Vrt(4) =  1.D0/(2.D0*Pi*Vr(4)*VLO(4))*VQ2
C...
C...   Inferior Vena Cava   [Venous Segment 3]
C...
        VQ2  = VQ(3)  - VOQ(3)
        VQ2t = VQt(3) - VOQt(3)
        VR2  = R2/VCAP(3)
        VPt(3) = VQ2t*VR2 + VQ2/VCAP(3)
        Vrt(3) = 1.D0/(2.D0*Pi*Vr(3)*VLO(3))*VQ2
C...
C...
C...  Above the heart
C...
C...   Subclavian - Upper Thorax   [Arterial Segment 15]
C...
      AP(15) = AP(3)
      AQ2 = AQ(15) - AQ(16)
      AQ2t = AQt(15) - AQt(16)
      AR2 = R2/ACAP(15)
      APt(15) = (AQ2t*AR2 + AQ2/ACAP(15) + APt(3))/2.D0
      AQt(15) =  1.D0/AINERT(2)*(AP(2) - AP(15)
     *          +  ADPG(2) - ARES(2)*(AQ(3) + AQ(15))) - AQt(3)
      AQt(16) = 1.D0/AINERT(15)*(AP(15) - AP(16) + ADPG(15)
     *          - ARES(15)*AQ(16))
      Art(15) =  1.D0/(2.D0*Pi*Ar(15)*ALO(15))*AQ2
      Art(15) = 0.D0
C...
C...  Lower Neck   [Arterial Segment 16]
C...
      AQ2 = AQ(16) - AQ(17)
      AQ2t = AQt(16) - AQt(17)
      AR2 = R2/ACAP(16)
      APt(16) = AQ2t*AR2 + AQ2/ACAP(16)
      AQt(17) =  1.D0/AINERT(16)*(AP(16) - AP(17)
     *       +  ADPG(16) - ARES(16)*AQ(17))
      Art(16) =  1.D0/(2.D0*Pi*Ar(16)*ALO(16))*AQ2
       Art(16) = 0.D0
C...
C...    Upper Neck (Carotid sinus)   [Arterial Segment 17]
C...
      AQ2 =  AQ(17) - AQ(18)
      AQ2t = AQt(17) - AQt(18)
      AR2 = R2/ACAP(17)
      APt(17) = AQ2t*AR2 + AQ2/ACAP(17)
      AQt(18) = 1.D0/AINERT(17)*(AP(17) - AP(18)
     *           + ADPG(17) - ARES(17)*AQ(18))
      Art(17) =  1.D0/(2.D0*Pi*Ar(17)*ALO(17))*AQ2
      Art(17) = 0.D0
```

```
C...
C...   Opthalmic   [Arterial Segment 18]
C...
      AQ2 = AQ(18) - (AQ(19) + PQ(18))
      AQ2t = AQt(18) - (AQt(19) + PQt(18))
      AR2 = R2/ACAP(18)
      APt(18) = AQ2t*AR2 + AQ2/ACAP(18)
      AQt(19) = 1.D0/AINERT(18)*(AP(18) - AP(19) + ADPG(18)
     *          - ARES(18)*(AQ(19) + PQ(18))) - PQt(18)
      Art(18) =  1.D0/(2.D0*Pi*Ar(18)*ALO(18))*AQ2
      Art(18) =  0.D0
C...
C...   Midbrain   [Arterial Segment 19]
C...
      AQ2 = AQ(19) - AQ(20)
      AQ2t = AQt(19) - AQt(20)
      AR2 = R2/ACAP(19)
      APt(19) = AQ2t*AR2 + AQ2/ACAP(19)
      AQt(20) = 1.D0/AINERT(19)*(AP(19) - AP(20) + ADPG(19)
     *          - ARES(19)*AQ(20))
      Art(19) =  1.D0/(2.D0*Pi*Ar(19)*ALO(19))*AQ2
      Art(19) = 0.D0
C...
C...   Cerebral  [Arterial Segment 20]
C...
      AQ2 = AQ(20) - PQ(20)
      AQ2t = AQt(20) - PQt(20)
      AR2 = R2/ACAP(20)
      APt(20) = AQ2t*AR2 + AQ2/ACAP(20)
      Art(20) =  1.D0/(2.D0*Pi*Ar(20)*ALO(20))*AQ2
      Art(20) = 0.D0
C...
C...  Cerebral bed   [Peripheral Segment 20]
C...
      PQt(20) = 1.D0/AINERT(20)*(AP(20) - PP(20) + ADPG(20)
     *          - ARES(20)*PQ(20))
      VQ(20) = DMAX1((PP(20) - VP(20) - PQ(20)*PRA(20))/PRV(20),0.D0)
      PQ2 = PQ(20) - VQ(20)
      PPt(20) = PQt(20)*PRA(20) + PQ2/PCAP(20)
C...
C...  Venous Drainage from Brain  [Venous Segment 20]
C...
      VQt(20) = (PPt(20) - VPt(20) - PQt(20)*PRA(20))/PRV(20)
      VQ2 = VQ(20) - VQ(19)
      VQ2t = VQt(20) - VQt(19)
      VR2 = R2/VCAP(20)
      VPt(20) = VQ2t*VR2 + VQ2/VCAP(20)
      VQt(19) = 1.D0/VINERT(20)*(VP(20) - VP(19) + VDPG(20)
     *          - VRES(20)*VQ(19))
      Vrt(20) =  1.D0/(2.D0*Pi*Vr(20)*VLO(20))*VQ2
C...
C...  Midbrain   [Venous Segment 19]
C...
      PQOUT(18) = (PP(18) - VP(18) - PRA(18)*PQ(18))/PRV(18)
      PQOUTt = (PPt(18) - VPt(18) - PRA(18)*PQt(18))/PRV(18)
      VQ2 =  VQ(19) - (VQ(18) - PQOUT(18))
      VQ2t = VQt(19) - (VQt(18) - PQOUTt)
      VR2  = R2/VCAP(19)
      VPt(19) = VQ2t*VR2 + VQ2/VCAP(19)
      VQt(18) = 1.D0/VINERT(19)*(VP(19) - VP(18) + VDPG(19)
     *          - VRES(19)*(VQ(18) - PQOUT(18))) + PQOUTt
      Vrt(19) =  1.D0/(2.D0*Pi*Vr(19)*VLO(19))*VQ2
C...
C...   Opthalmic   [Venous Segment 18]
C...
      VQ2 =  VQ(18) - VQ(17)
      VQ2t = VQt(18) - VQt(17)
      VR2 = R2/VCAP(18)
      VPt(18) = VQ2t*VR2 + VQ2/VCAP(18)
      VQt(17) = 1.D0/VINERT(18)*(VP(18) - VP(17) + VDPG(18)
     *             - VRES(18)*VQ(17))
      Vrt(18) =  1.D0/(2.D0*Pi*Vr(18)*VLO(18))*VQ2
C...
C...   Upper Neck   [Venous Segment 17]
C...
      VQ2 =  VQ(17) - VQ(16)
      VQ2t = VQt(17) - VQt(16)
      VR2 = R2/VCAP(17)
      VPt(17) = VQ2t*VR2 + VQ2/VCAP(17)
      VQt(16) = 1.D0/VINERT(17)*(VP(17) - VP(16) + VDPG(17)
     *             - VRES(17)*VQ(16))
      Vrt(17) =  1.D0/(2.D0*Pi*Vr(17)*VLO(17))*VQ2
C...
C...   Lower Neck (Jugler)   [Venous Segment 16]
C...
      VQ2 =  VQ(16) - VQ(15)
      VQ2t = VQt(16) - VQt(15)
      VR2 = R2/VCAP(16)
      VPt(16) = VQ2t*VR2 + VQ2/VCAP(16)
      VQt(15) = 1.D0/VINERT(16)*(VP(16) - VP(15) + VDPG(16)
     *             - VRES(16)*VQ(15))
      Vrt(16) =  1.D0/(2.D0*Pi*Vr(16)*VLO(16))*VQ2
C...
C...   Subclavian   [Venous Segment 15]
C...
      VQ2 =  VQ(15) - VQ(2)
      VQ2t = VQt(15) - VQt(2)
      VR2 = R2/VCAP(15)
      VPt(15) = VQ2t*VR2 + VQ2/VCAP(15)
      VQt(2) = 1.D0/VINERT(15)*(VP(15) - VP(2) + VDPG(15)
     *             - VRES(15)*VQ(2))
      Vrt(15) =  1.D0/(2.D0*Pi*Vr(15)*VLO(15))*VQ2
C...
C...  Superior Vena Cava  [Venous Segment 2]
C...
        VQ2 =  VQ(2)  - VOQ(2)
        VQ2t = VQt(2) - VOQt(2)
        VR2 = R2/VCAP(2)
        VPt(2) = VQ2t*VR2 + VQ2/VCAP(2)
        Vrt(2) =  1.D0/(2.D0*Pi*Vr(2)*VLO(2))*VQ2
C...
C...  Form the derivatives for the peripheral beds
C...
C...   Opthalmic   [Peripheral Segment 18]
C...
      PQ2  = PQ(18) - PQOUT(18)
      PQt(18) = PPt(18)/PRA(18) - PQ2/(PRA(18)*PCAP(18))
C...
C...  Thorax and Coronaries [Peripheral Segment 3]
C...
      PQ2  = PQ(3)  - PQOUT(3)
      PQt(3) =  PPt(3)/PRA(3) - PQ2/(PRA(3)*PCAP(3))
C...
C...   Remainder of Peripheral Segments
C...
      i = 6
      DO WHILE(i .LE. IFOOTSEG - 1)
         IF ( POUT(i) .GT. 0 ) THEN
            PQ2  = PQ(i) - PQOUT(i)
            PQt(i) =  PPt(i)/PRA(i) - PQ2/(PRA(i)*PCAP(i))
         END IF
         i = i + 1
      END DO
      RETURN
      END
      SUBROUTINE PRINT(NI,NO6, NO7, NO8)
C...
C...  ODE COMMON
```

```
C...
C...   /Y/ time variables
C...   /F/ time derivatives of variables
C...   /S/ spatial derivatives of variables
C...   /R/ & /I/ real and integer parameters required to define constants and
C...            define the spatial integration grid.
C...
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      PARAMETER (NEQ = 20, NPSEG = 10)
      INTEGER NSTOP, NORUN, IP
      INTEGER*2 ALIN, ALOUT, VLIN, VLOUT, PVS, PIN, POUT
      DOUBLE PRECISION MM2PA, mu0
      COMMON/T/         T,      NSTOP,     NORUN    ! Run Parameters
C...
C...   Arrays for segmental variables Pressure (P), Flow(Q), and radii (r)
C...
     1  /Y/    HP(4),      HQ(4),                     ! Heart's Chambers RA=1
     *        AP(NEQ),   AQ(NEQ),   Ar(NEQ),      ! Arterial P, Q, r
     *        VP(NEQ),   VQ(NEQ),   Vr(NEQ),      ! Venous P, Q, r
     *        VOP(3),     VOQ(3),                     ! Venous flows into heart
     *        PP(NEQ),   PQ(NEQ),                     ! Peripheral P, Qin, Qout
C...
C...  Time derivatives of the segmental variables: Pt, Qt, rt
C...
     2  /F/   HPt(4),     HQt(4),                         ! Heart's Chambers RA=1
     *       APt(NEQ),  AQt(NEQ),   Art(NEQ),         ! Arterial Pt, Qt, rt
     *       VPt(NEQ),  VQt(NEQ),   Vrt(NEQ),         ! Venous Pt, Qt, rt
     *        VOPt(3),   VOQt(3),                         ! Venous flows into heart
     *       PPt(NEQ),  PQt(NEQ),                         ! Peripheral Pt, QINt, QOUTt
C...
C... Parameters necessary to form the differential equations
C...
C...    Pi  -  3.14159...
C...    g0  -  9.80665 [m/sec^2]  earth's acceleration of gravity
C...  rho0  -  1050.   [kg/m^3]   density of whole blood (45% Hct)  | Assumed
C...   mu0  -  2.7     [cp]       viscosity of whole blood (45% Hct)| Constant
C...   D2R  -  pi/180  [radians/degree] scale factor
C...
     3  /R/  Pi, g0, rho0, mu0,  D2R, MM2PA, R2,        ! Constants
     *       ZAO(NEQ), ZAT(NEQ), ZVO(NEQ), ZVT(NEQ),   ! Arterial & Venous
     *       THETA(NEQ), HCAP(4), HVOL(4), HVR(4),                               ! Orientation angle
     *       ALO(NEQ), ArU(NEQ),  AE(NEQ), Ah(NEQ),    ! Arterial l, r, E, h
     *       ACAP(NEQ), ARES(NEQ), AINERT(NEQ),        ! Arterial Capacitance, resistance
     *       VLO(NEQ),  VrO(NEQ),  VrU(NEQ),           ! Venous l, r, rUNSTRESSED
     *       VCAPO(NEQ), VCAP(NEQ), VRES(NEQ), VINERT(NEQ), ! Venous Capacitance, resistance, inertance
     *       PRA(NEQ),  PRV(NEQ), PCAP(NEQ),           ! Peripheral Ra, Rv, C
     *       PINERT(NEQ), PVOL(NEQ),                   ! Peripheral I, V
     *       PAO(NEQ),  PVO(NEQ),                      ! Initial P conditions
     *       QAO(NEQ),  QVO(NEQ),                      ! Initial Q conditions
     *       AVOL(NEQ), VVOL(NEQ),                     ! A & V Volumes
     *       PEXT(NEQ),                                ! Externally applied Pressure
     *       TO, GSTART, GMAX, TBRK1, TBRK2, TMAX, GFIN, ! G-Profile parameters
     *       Gz, ADPG(NEQ), VDPG(NEQ)                  ! Gz & Delta P from G
     4  /I/  IP, NDXPER(NPSEG),                        ! Peripheral Bed Indexes
     *       ALIN(NEQ), ALOUT(NEQ),                    ! Linkage Data arterial
     *       VLIN(NEQ), VLOUT(NEQ),                    ! Venous
     *       PIN(NEQ), POUT(NEQ),                    ! Peripheral
     *       PVS(NEQ),                                 ! Number parallel venous segments
     *       IFOOTSEG, IHEADSEG, IHEARTSEG             ! Foot, Head, & Heart seg nums
C...
C...
C...  PRINT A HEADING FOR THE NUMERICAL SOLUTION
      IP=IP+1
      IF(IP.EQ.1)WRITE(NO6,100)
      IF(IP.EQ.1)WRITE(NO7,100)
      IF(IP.EQ.1)WRITE(NO8,100)
      IF(IP.EQ.1)WRITE(*,1)
C      WRITE(NO,2)
C      WRITE(NO,2) T
```

```
    1  FORMAT(2X,'T = ',F12.4,' sec'  )
C...
C...  PRINT THE SOLUTION
C...
       HQ(1) = VOQ(2) + VOQ(3)
       WRITE( *,22) T, Gz, THETA(1),(VP(k), k=1,NEQ), (VQ(k), k=1,NEQ)
       WRITE( *,22)  T, Gz, THETA(1),(HP(k), k=1,4), (HQ(k), k=1,4)
       WRITE(NO6,2) T, Gz, THETA(1),(AP(k), k=1,NEQ), (AQ(k), k=1,NEQ)
       WRITE(NO7,2) T, Gz, THETA(1),(VP(k), k=1,NEQ), (VQ(k), k=1,NEQ)
       WRITE(NO8,2) T, Gz, THETA(1),(PP(k), k=1,NEQ), (PQ(k), k=1,NEQ)
    2  FORMAT(3F8.4,60(E12.4,1X))
   22  FORMAT(3F8.4, /,12(5(E12.4,1X),1X,/))
  100  FORMAT(2X,' Time',3X,'     Gz     ','   Theta    ','Op Art Press',
     *' Op Art Flow',' Op Vn Press',' Op Vn Flow','Op PBed Pr  ')
       RETURN
       END
```

PROGRAM RKF45.FOR

```
*DECK RKF45
      SUBROUTINE RKF45(F,NEQN,Y,T,TOUT,RELERR,ABSERR,IFLAG,WORK,IWORK)
C
C     FEHLBERG FOURTH-FIFTH ORDER RUNGE-KUTTA METHOD
C
C     WRITTEN BY H.A.WATTS AND L.F.SHAMPINE
C                   SANDIA LABORATORIES
C                  ALBUQUERQUE,NEW MEXICO
C
C    RKF45 IS PRIMARILY DESIGNED TO SOLVE NON-STIFF AND MILDLY STIFF
C    DIFFERENTIAL EQUATIONS WHEN DERIVATIVE EVALUATIONS ARE INEXPENSIVE.
C    RKF45 SHOULD GENERALLY NOT BE USED WHEN THE USER IS DEMANDING
C    HIGH ACCURACY.
C
C ABSTRACT
C
C    SUBROUTINE  RKF45  INTEGRATES A SYSTEM OF NEQN FIRST ORDER
C    ORDINARY DIFFERENTIAL EQUATIONS OF THE FORM
C             DY(I)/DT = F(T,Y(1),Y(2),...,Y(NEQN))
C              WHERE THE Y(I) ARE GIVEN AT T .
C    TYPICALLY THE SUBROUTINE IS USED TO INTEGRATE FROM T TO TOUT BUT IT
C    CAN BE USED AS A ONE-STEP INTEGRATOR TO ADVANCE THE SOLUTION A
C    SINGLE STEP IN THE DIRECTION OF TOUT.  ON RETURN THE PARAMETERS IN
C    THE CALL LIST ARE SET FOR CONTINUING THE INTEGRATION. THE USER HAS
C    ONLY TO CALL RKF45 AGAIN (AND PERHAPS DEFINE A NEW VALUE FOR TOUT).
C    ACTUALLY, RKF45 IS AN INTERFACING ROUTINE WHICH CALLS SUBROUTINE
C    RKFS FOR THE SOLUTION.  RKFS IN TURN CALLS SUBROUTINE  FEHL WHICH
C    COMPUTES AN APPROXIMATE SOLUTION OVER ONE STEP.
C
C    RKF45  USES THE RUNGE-KUTTA-FEHLBERG (4,5)  METHOD DESCRIBED
C    IN THE REFERENCE
C    E.FEHLBERG , LOW-ORDER CLASSICAL RUNGE-KUTTA FORMULAS WITH STEPSIZE
C                 CONTROL , NASA TR R-315
C
C    THE PERFORMANCE OF RKF45 IS ILLUSTRATED IN THE REFERENCE
C    L.F.SHAMPINE,H.A.WATTS,S.DAVENPORT, SOLVING NON-STIFF ORDINARY
C                 DIFFERENTIAL EQUATIONS-THE STATE OF THE ART ,
C                 SANDIA LABORATORIES REPORT SAND75-0182 ,
C                 TO APPEAR IN SIAM REVIEW.
C
C
C    THE PARAMETERS REPRESENT-
C      F -- SUBROUTINE F(T,Y,YP) TO EVALUATE DERIVATIVES YP(I)=DY(I)/DT
C      NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED
C      Y(*) -- SOLUTION VECTOR AT T
C      T -- INDEPENDENT VARIABLE
C      TOUT -- OUTPUT POINT AT WHICH SOLUTION IS DESIRED
C      RELERR,ABSERR -- RELATIVE AND ABSOLUTE ERROR TOLERANCES FOR LOCAL
C            ERROR TEST. AT EACH STEP THE CODE REQUIRES THAT
C                 ABS(LOCAL ERROR) .LE. RELERR*ABS(Y) + ABSERR
C            FOR EACH COMPONENT OF THE LOCAL ERROR AND SOLUTION VECTORS
C      IFLAG -- INDICATOR FOR STATUS OF INTEGRATION
C      WORK(*) -- ARRAY TO HOLD INFORMATION INTERNAL TO RKF45 WHICH IS
C            NECESSARY FOR SUBSEQUENT CALLS. MUST BE DIMENSIONED
C            AT LEAST  3+6*NEQN
C      IWORK(*) -- INTEGER ARRAY USED TO HOLD INFORMATION INTERNAL TO
C            RKF45 WHICH IS NECESSARY FOR SUBSEQUENT CALLS. MUST BE
C            DIMENSIONED AT LEAST  5
C
C
C  FIRST CALL TO RKF45
C
C    THE USER MUST PROVIDE STORAGE IN HIS CALLING PROGRAM FOR THE ARRAYS
C    IN THE CALL LIST  -      Y(NEQN) , WORK(3+6*NEQN) , IWORK(5)  ,
C    DECLARE F IN AN EXTERNAL STATEMENT, SUPPLY SUBROUTINE F(T,Y,YP) AND
C    INITIALIZE THE FOLLOWING PARAMETERS-
C
```

```
C       NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED.  (NEQN .GE. 1)
C       Y(*) -- VECTOR OF INITIAL CONDITIONS
C       T -- STARTING POINT OF INTEGRATION , MUST BE A VARIABLE
C       TOUT -- OUTPUT POINT AT WHICH SOLUTION IS DESIRED.
C             T=TOUT IS ALLOWED ON THE FIRST CALL ONLY, IN WHICH CASE
C             RKF45 RETURNS WITH IFLAG=2 IF CONTINUATION IS POSSIBLE.
C       RELERR,ABSERR -- RELATIVE AND ABSOLUTE LOCAL ERROR TOLERANCES
C             WHICH MUST BE NON-NEGATIVE. RELERR MUST BE A VARIABLE WHILE
C             ABSERR MAY BE A CONSTANT. THE CODE SHOULD NORMALLY NOT BE
C             USED WITH RELATIVE ERROR CONTROL SMALLER THAN ABOUT 1.E-8 .
C             TO AVOID LIMITING PRECISION DIFFICULTIES THE CODE REQUIRES
C             RELERR TO BE LARGER THAN AN INTERNALLY COMPUTED RELATIVE
C             ERROR PARAMETER WHICH IS MACHINE DEPENDENT. IN PARTICULAR,
C             PURE ABSOLUTE ERROR IS NOT PERMITTED. IF A SMALLER THAN
C             ALLOWABLE VALUE OF RELERR IS ATTEMPTED, RKF45 INCREASES
C             RELERR APPROPRIATELY AND RETURNS CONTROL TO THE USER BEFORE
C             CONTINUING THE INTEGRATION.
C       IFLAG -- +1,-1  INDICATOR TO INITIALIZE THE CODE FOR EACH NEW
C             PROBLEM. NORMAL INPUT IS +1. THE USER SHOULD SET IFLAG=-1
C             ONLY WHEN ONE-STEP INTEGRATOR CONTROL IS ESSENTIAL. IN THIS
C             CASE, RKF45 ATTEMPTS TO ADVANCE THE SOLUTION A SINGLE STEP
C             IN THE DIRECTION OF TOUT EACH TIME IT IS CALLED. SINCE THIS
C             MODE OF OPERATION RESULTS IN EXTRA COMPUTING OVERHEAD, IT
C             SHOULD BE AVOIDED UNLESS NEEDED.
C
C
C   OUTPUT FROM RKF45
C
C       Y(*) -- SOLUTION AT T
C       T -- LAST POINT REACHED IN INTEGRATION.
C       IFLAG = 2 -- INTEGRATION REACHED TOUT. INDICATES SUCCESSFUL RETURN
C                    AND IS THE NORMAL MODE FOR CONTINUING INTEGRATION.
C             =-2 -- A SINGLE SUCCESSFUL STEP IN THE DIRECTION OF TOUT
C                    HAS BEEN TAKEN. NORMAL MODE FOR CONTINUING
C                    INTEGRATION ONE STEP AT A TIME.
C             = 3 -- INTEGRATION WAS NOT COMPLETED BECAUSE RELATIVE ERROR
C                    TOLERANCE WAS TOO SMALL. RELERR HAS BEEN INCREASED
C                    APPROPRIATELY FOR CONTINUING.
C             = 4 -- INTEGRATION WAS NOT COMPLETED BECAUSE MORE THAN
C                    3000 DERIVATIVE EVALUATIONS WERE NEEDED. THIS
C                    IS APPROXIMATELY 500 STEPS.
C             = 5 -- INTEGRATION WAS NOT COMPLETED BECAUSE SOLUTION
C                    VANISHED MAKING A PURE RELATIVE ERROR TEST
C                    IMPOSSIBLE. MUST USE NON-ZERO ABSERR TO CONTINUE.
C                    USING THE ONE-STEP INTEGRATION MODE FOR ONE STEP
C                    IS A GOOD WAY TO PROCEED.
C             = 6 -- INTEGRATION WAS NOT COMPLETED BECAUSE REQUESTED
C                    ACCURACY COULD NOT BE ACHIEVED USING SMALLEST
C                    ALLOWABLE STEPSIZE. USER MUST INCREASE THE ERROR
C                    TOLERANCE BEFORE CONTINUED INTEGRATION CAN BE
C                    ATTEMPTED.
C             = 7 -- IT IS LIKELY THAT RKF45 IS INEFFICIENT FOR SOLVING
C                    THIS PROBLEM. TOO MUCH OUTPUT IS RESTRICTING THE
C                    NATURAL STEPSIZE CHOICE. USE THE ONE-STEP INTEGRATOR
C                    MODE.
C             = 8 -- INVALID INPUT PARAMETERS
C                    THIS INDICATOR OCCURS IF ANY OF THE FOLLOWING IS
C                    SATISFIED -   NEQN .LE. 0
C                                  T=TOUT  AND  IFLAG .NE. +1 OR -1
C                                  RELERR OR ABSERR .LT. 0.
C                                  IFLAG .EQ. 0  OR  .LT. -2  OR  .GT. 8
C       WORK(*),IWORK(*) -- INFORMATION WHICH IS USUALLY OF NO INTEREST
C                    TO THE USER BUT NECESSARY FOR SUBSEQUENT CALLS.
C                    WORK(1),...,WORK(NEQN) CONTAIN THE FIRST DERIVATIVES
C                    OF THE SOLUTION VECTOR Y AT T. WORK(NEQN+1) CONTAINS
C                    THE STEPSIZE H TO BE ATTEMPTED ON THE NEXT STEP.
C                    IWORK(1) CONTAINS THE DERIVATIVE EVALUATION COUNTER.
C
C
```

```
C  SUBSEQUENT CALLS TO RKF45
C
C     SUBROUTINE RKF45 RETURNS WITH ALL INFORMATION NEEDED TO CONTINUE
C     THE INTEGRATION. IF THE INTEGRATION REACHED TOUT, THE USER NEED ONL
C     DEFINE A NEW TOUT AND CALL RKF45 AGAIN. IN THE ONE-STEP INTEGRATOR
C     MODE (IFLAG=-2) THE USER MUST KEEP IN MIND THAT EACH STEP TAKEN IS
C     IN THE DIRECTION OF THE CURRENT TOUT. UPON REACHING TOUT (INDICATED
C     BY CHANGING IFLAG TO 2),THE USER MUST THEN DEFINE A NEW TOUT AND
C     RESET IFLAG TO -2 TO CONTINUE IN THE ONE-STEP INTEGRATOR MODE.
C
C     IF THE INTEGRATION WAS NOT COMPLETED BUT THE USER STILL WANTS TO
C     CONTINUE (IFLAG=3,4 CASES), HE JUST CALLS RKF45 AGAIN. WITH IFLAG=3
C     THE RELERR PARAMETER HAS BEEN ADJUSTED APPROPRIATELY FOR CONTINUING
C     THE INTEGRATION. IN THE CASE OF IFLAG=4 THE FUNCTION COUNTER WILL
C     BE RESET TO 0 AND ANOTHER 3000 FUNCTION EVALUATIONS ARE ALLOWED.
C
C     HOWEVER,IN THE CASE IFLAG=5, THE USER MUST FIRST ALTER THE ERROR
C     CRITERION TO USE A POSITIVE VALUE OF ABSERR BEFORE INTEGRATION CAN
C     PROCEED. IF HE DOES NOT,EXECUTION IS TERMINATED.
C
C     ALSO,IN THE CASE IFLAG=6, IT IS NECESSARY FOR THE USER TO RESET
C     IFLAG TO 2 (OR -2 WHEN THE ONE-STEP INTEGRATION MODE IS BEING USED)
C     AS WELL AS INCREASING EITHER ABSERR,RELERR OR BOTH BEFORE THE
C     INTEGRATION CAN BE CONTINUED. IF THIS IS NOT DONE, EXECUTION WILL
C     BE TERMINATED. THE OCCURRENCE OF IFLAG=6 INDICATES A TROUBLE SPOT
C     (SOLUTION IS CHANGING RAPIDLY,SINGULARITY MAY BE PRESENT) AND IT
C     OFTEN IS INADVISABLE TO CONTINUE.
C
C     IF IFLAG=7 IS ENCOUNTERED, THE USER SHOULD USE THE ONE-STEP
C     INTEGRATION MODE WITH THE STEPSIZE DETERMINED BY THE CODE OR
C     CONSIDER SWITCHING TO THE ADAMS CODES DE/STEP,INTRP. IF THE USER
C     INSISTS UPON CONTINUING THE INTEGRATION WITH RKF45, HE MUST RESET
C     IFLAG TO 2 BEFORE CALLING RKF45 AGAIN. OTHERWISE,EXECUTION WILL BE
C     TERMINATED.
C
C     IF IFLAG=8 IS OBTAINED, INTEGRATION CAN NOT BE CONTINUED UNLESS
C     THE INVALID INPUT PARAMETERS ARE CORRECTED.
C
C     IT SHOULD BE NOTED THAT THE ARRAYS WORK,IWORK CONTAIN INFORMATION
C     REQUIRED FOR SUBSEQUENT INTEGRATION. ACCORDINGLY, WORK AND IWORK
C     SHOULD NOT BE ALTERED.
C
C
      INTEGER NEQN,IFLAG,IWORK(5)
      DOUBLE PRECISION Y(NEQN),T,TOUT,RELERR,ABSERR,WORK(3500)
C     IF COMPILER CHECKS SUBSCRIPTS, CHANGE WORK(1) TO WORK(3+6*NEQN)
C
      EXTERNAL F
C
      INTEGER K1,K2,K3,K4,K5,K6,K1M
C
C
C     COMPUTE INDICES FOR THE SPLITTING OF THE WORK ARRAY
C
      K1M=NEQN+1
      K1=K1M+1
      K2=K1+NEQN
      K3=K2+NEQN
      K4=K3+NEQN
      K5=K4+NEQN
      K6=K5+NEQN
C
C     THIS INTERFACING ROUTINE MERELY RELIEVES THE USER OF A LONG
C     CALLING LIST VIA THE SPLITTING APART OF TWO WORKING STORAGE
C     ARRAYS. IF THIS IS NOT COMPATIBLE WITH THE USERS COMPILER,
C     HE MUST USE RKFS DIRECTLY.
C
      CALL RKFS(F,NEQN,Y,T,TOUT,RELERR,ABSERR,IFLAG,WORK(1),WORK(K1M),
     1          WORK(K1),WORK(K2),WORK(K3),WORK(K4),WORK(K5),WORK(K6),
     2          WORK(K6+1),IWORK(1),IWORK(2),IWORK(3),IWORK(4),IWORK(5))
C
      RETURN
      END
      SUBROUTINE RKFS(F,NEQN,Y,T,TOUT,RELERR,ABSERR,IFLAG,YP,H,F1,F2,F3,
     1               F4,F5,SAVRE,SAVAE,NFE,KOP,INIT,JFLAG,KFLAG)
C
C     FEHLBERG FOURTH-FIFTH ORDER RUNGE-KUTTA METHOD
C
C
C     RKFS INTEGRATES A SYSTEM OF FIRST ORDER ORDINARY DIFFERENTIAL
C     EQUATIONS AS DESCRIBED IN THE COMMENTS FOR RKF45 .
C     THE ARRAYS YP,F1,F2,F3,F4,AND F5 (OF DIMENSION AT LEAST NEQN) AND
C     THE VARIABLES H,SAVRE,SAVAE,NFE,KOP,INIT,JFLAG,AND KFLAG ARE USED
C     INTERNALLY BY THE CODE AND APPEAR IN THE CALL LIST TO ELIMINATE
C     LOCAL RETENTION OF VARIABLES BETWEEN CALLS. ACCORDINGLY, THEY
C     SHOULD NOT BE ALTERED. ITEMS OF POSSIBLE INTEREST ARE
C         YP - DERIVATIVE OF SOLUTION VECTOR AT T
C         H  - AN APPROPRIATE STEPSIZE TO BE USED FOR THE NEXT STEP
C         NFE- COUNTER ON THE NUMBER OF DERIVATIVE FUNCTION EVALUATIONS
C
C
      LOGICAL HFAILD,OUTPUT
C
      INTEGER  NEQN,IFLAG,NFE,KOP,INIT,JFLAG,KFLAG
      DOUBLE PRECISION  Y(NEQN),T,TOUT,RELERR,ABSERR,H,YP(NEQN),
     1  F1(NEQN),F2(NEQN),F3(NEQN),F4(NEQN),F5(NEQN),SAVRE,
     2  SAVAE
C
      EXTERNAL F
C
      DOUBLE PRECISION  A,AE,DT,EE,EEOET,ESTTOL,ET,HMIN,REMIN,RER,S,
     1  SCALE,TOL,TOLN,U26,EPSP1,EPS,YPK
C
      INTEGER  K,MAXNFE,MFLAG
C
      DOUBLE PRECISION  DABS,DMAX1,DMIN1,DSIGN
C
C     REMIN IS THE MINIMUM ACCEPTABLE VALUE OF RELERR.  ATTEMPTS
C     TO OBTAIN HIGHER ACCURACY WITH THIS SUBROUTINE ARE USUALLY
C     VERY EXPENSIVE AND OFTEN UNSUCCESSFUL.
C
      DATA REMIN/1.0D-12/
C
C
C     THE EXPENSE IS CONTROLLED BY RESTRICTING THE NUMBER
C     OF FUNCTION EVALUATIONS TO BE APPROXIMATELY MAXNFE.
C     AS SET, THIS CORRESPONDS TO ABOUT 500 STEPS.
C
      DATA MAXNFE/2000000/
C
C
C     CHECK INPUT PARAMETERS
C
C
      IF (NEQN .LT. 1) GO TO 10
      IF ((RELERR .LT. 0.0D0)  .OR.  (ABSERR .LT. 0.0D0)) GO TO 10
      MFLAG=IABS(IFLAG)
      IF ((MFLAG .EQ. 0) .OR. (MFLAG .GT. 8)) GO TO 10
      IF (MFLAG .NE. 1) GO TO 20
C
C     FIRST CALL, COMPUTE MACHINE EPSILON
C
      EPS = 1.0D0
    5 EPS = EPS/2.0D0
      EPSP1 = EPS + 1.0D0
      IF (EPSP1 .GT. 1.0D0) GO TO 5
      U26 = 26.0D0*EPS
      GO TO 50
```

```
C
C     INVALID INPUT
   10 IFLAG=8
      RETURN
C
C     CHECK CONTINUATION POSSIBILITIES
C
   20 IF ((T .EQ. TOUT) .AND. (KFLAG .NE. 3)) GO TO 10
      IF (MFLAG .NE. 2) GO TO 25
C
C     IFLAG = +2 OR -2
      IF ((KFLAG .EQ. 3) .OR. (INIT .EQ. 0)) GO TO 45
      IF (KFLAG .EQ. 4) GO TO 40
      IF ((KFLAG .EQ. 5)  .AND.  (ABSERR .EQ. 0.0D0)) GO TO 30
      IF ((KFLAG .EQ. 6)  .AND.  (RELERR .LE. SAVRE)  .AND.
     1    (ABSERR .LE. SAVAE)) GO TO 30
      GO TO 50
C
C     IFLAG = 3,4,5,6,7 OR 8
   25 IF (IFLAG .EQ. 3) GO TO 45
      IF (IFLAG .EQ. 4) GO TO 40
      IF ((IFLAG .EQ. 5) .AND. (ABSERR .GT. 0.0D0)) GO TO 45
C
C     INTEGRATION CANNOT BE CONTINUED SINCE USER DID NOT RESPOND TO
C     THE INSTRUCTIONS PERTAINING TO IFLAG=5,6,7 OR 8
   30 STOP
C
C     RESET FUNCTION EVALUATION COUNTER
   40 NFE=0
      IF (MFLAG .EQ. 2) GO TO 50
C
C     RESET FLAG VALUE FROM PREVIOUS CALL
   45 IFLAG=JFLAG
      IF (KFLAG .EQ. 3) MFLAG=IABS(IFLAG)
C
C     SAVE INPUT IFLAG AND SET CONTINUATION FLAG VALUE FOR SUBSEQUENT
C     INPUT CHECKING
   50 JFLAG=IFLAG
      KFLAG=0
C
C     SAVE RELERR AND ABSERR FOR CHECKING INPUT ON SUBSEQUENT CALLS
      SAVRE=RELERR
      SAVAE=ABSERR
C
C     RESTRICT RELATIVE ERROR TOLERANCE TO BE AT LEAST AS LARGE AS
C     2*EPS+REMIN TO AVOID LIMITING PRECISION DIFFICULTIES ARISING
C     FROM IMPOSSIBLE ACCURACY REQUESTS
C
      RER=2.0D0*EPS+REMIN
      IF (RELERR .GE. RER) GO TO 55
C
C     RELATIVE ERROR TOLERANCE TOO SMALL
      RELERR=RER
      IFLAG=3
      KFLAG=3
      RETURN
C
   55 DT=TOUT-T
C
      IF (MFLAG .EQ. 1) GO TO 60
      IF (INIT .EQ. 0) GO TO 65
      GO TO 80
C
C     INITIALIZATION --
C                       SET INITIALIZATION COMPLETION INDICATOR,INIT
C                       SET INDICATOR FOR TOO MANY OUTPUT POINTS,KOP
C                       EVALUATE INITIAL DERIVATIVES
C                       SET COUNTER FOR FUNCTION EVALUATIONS,NFE
C                       ESTIMATE STARTING STEPSIZE
```

```
C
   60 INIT=0
      KOP=0
C
      A=T
      CALL F(A,Y,YP)
      NFE=1
      IF (T .NE. TOUT) GO TO 65
      IFLAG=2
      RETURN
C
C
   65 INIT=1
      H=DABS(DT)
      TOLN=0.
      DO 70 K=1,NEQN
        TOL=RELERR*DABS(Y(K))+ABSERR
        IF (TOL .LE. 0.) GO TO 70
        TOLN=TOL
        YPK=DABS(YP(K))
        IF (YPK*H**5 .GT. TOL) H=(TOL/YPK)**0.2D0
   70 CONTINUE
      IF (TOLN .LE. 0.0D0) H=0.0D0
      H=DMAX1(H,U26*DMAX1(DABS(T),DABS(DT)))
      JFLAG=ISIGN(2,IFLAG)
C
C
C     SET STEPSIZE FOR INTEGRATION IN THE DIRECTION FROM T TO TOUT
C
   80 H=DSIGN(H,DT)
C
C     TEST TO SEE IF RKF45 IS BEING SEVERELY IMPACTED BY TOO MANY
C     OUTPUT POINTS
C
      IF (DABS(H) .GE. 2.0D0*DABS(DT)) KOP=KOP+1
      IF (KOP .NE. 100) GO TO 85
C
C     UNNECESSARY FREQUENCY OF OUTPUT
      KOP=0
      IFLAG=7
      RETURN
C
   85 IF (DABS(DT) .GT. U26*DABS(T)) GO TO 95
C
C     IF TOO CLOSE TO OUTPUT POINT,EXTRAPOLATE AND RETURN
C
      DO 90 K=1,NEQN
   90   Y(K)=Y(K)+DT*YP(K)
      A=TOUT
      CALL F(A,Y,YP)
      NFE=NFE+1
      GO TO 300
C
C
C     INITIALIZE OUTPUT POINT INDICATOR
C
   95 OUTPUT= .FALSE.
C
C     TO AVOID PREMATURE UNDERFLOW IN THE ERROR TOLERANCE FUNCTION,
C     SCALE THE ERROR TOLERANCES
C
      SCALE=2.0D0/RELERR
      AE=SCALE*ABSERR
C
C
C     STEP BY STEP INTEGRATION
C
  100 HFAILD= .FALSE.
C
```

```
C     SET SMALLEST ALLOWABLE STEPSIZE
C
      HMIN=U26*DABS(T)
C
C     ADJUST STEPSIZE IF NECESSARY TO HIT THE OUTPUT POINT.
C     LOOK AHEAD TWO STEPS TO AVOID DRASTIC CHANGES IN THE STEPSIZE AND
C     THUS LESSEN THE IMPACT OF OUTPUT POINTS ON THE CODE.
C
      DT=TOUT-T
      IF (DABS(DT) .GE. 2.0D0*DABS(H)) GO TO 200
      IF (DABS(DT) .GT. DABS(H)) GO TO 150
C
C     THE NEXT SUCCESSFUL STEP WILL COMPLETE THE INTEGRATION TO THE
C     OUTPUT POINT
C
      OUTPUT= .TRUE.
      H=DT
      GO TO 200
C
  150 H=0.5D0*DT
C
C
C
C     CORE INTEGRATOR FOR TAKING A SINGLE STEP
C
C     THE TOLERANCES HAVE BEEN SCALED TO AVOID PREMATURE UNDERFLOW IN
C     COMPUTING THE ERROR TOLERANCE FUNCTION ET.
C     TO AVOID PROBLEMS WITH ZERO CROSSINGS,RELATIVE ERROR IS MEASURED
C     USING THE AVERAGE OF THE MAGNITUDES OF THE SOLUTION AT THE
C     BEGINNING AND END OF A STEP.
C     THE ERROR ESTIMATE FORMULA HAS BEEN GROUPED TO CONTROL LOSS OF
C     SIGNIFICANCE.
C     TO DISTINGUISH THE VARIOUS ARGUMENTS, H IS NOT PERMITTED
C     TO BECOME SMALLER THAN 26 UNITS OF ROUNDOFF IN T.
C     PRACTICAL LIMITS ON THE CHANGE IN THE STEPSIZE ARE ENFORCED TO
C     SMOOTH THE STEPSIZE SELECTION PROCESS AND TO AVOID EXCESSIVE
C     CHATTERING ON PROBLEMS HAVING DISCONTINUITIES.
C     TO PREVENT UNNECESSARY FAILURES, THE CODE USES 9/10 THE STEPSIZE
C     IT ESTIMATES WILL SUCCEED.
C     AFTER A STEP FAILURE, THE STEPSIZE IS NOT ALLOWED TO INCREASE FOR
C     THE NEXT ATTEMPTED STEP. THIS MAKES THE CODE MORE EFFICIENT ON
C     PROBLEMS HAVING DISCONTINUITIES AND MORE EFFECTIVE IN GENERAL
C     SINCE LOCAL EXTRAPOLATION IS BEING USED AND EXTRA CAUTION SEEMS
C     WARRANTED.
C
C
C     TEST NUMBER OF DERIVATIVE FUNCTION EVALUATIONS.
C     IF OKAY,TRY TO ADVANCE THE INTEGRATION FROM T TO T+H
C
  200 IF (NFE .LE. MAXNFE) GO TO 220
C
C     TOO MUCH WORK
      IFLAG=4
      KFLAG=4
      RETURN
C
C     ADVANCE AN APPROXIMATE SOLUTION OVER ONE STEP OF LENGTH H
C
  220 CALL FEHL(F,NEQN,Y,T,H,YP,F1,F2,F3,F4,F5,F1)
      NFE=NFE+5
C
C     COMPUTE AND TEST ALLOWABLE TOLERANCES VERSUS LOCAL ERROR ESTIMATES
C     AND REMOVE SCALING OF TOLERANCES. NOTE THAT RELATIVE ERROR IS
C     MEASURED WITH RESPECT TO THE AVERAGE OF THE MAGNITUDES OF THE
C     SOLUTION AT THE BEGINNING AND END OF THE STEP.
C
      EEOET=0.0D0
      DO 250 K=1,NEQN
        ET=DABS(Y(K))+DABS(F1(K))+AE
        IF (ET .GT. 0.0D0) GO TO 240
C
C       INAPPROPRIATE ERROR TOLERANCE
        IFLAG=5
        RETURN
C
  240   EE=DABS((-2090.0D0*YP(K)+(21970.0D0*F3(K)-15048.0D0*F4(K)))+
     1                         (22528.0D0*F2(K)-27360.0D0*F5(K)))
  250   EEOET=DMAX1(EEOET,EE/ET)
C
      ESTTOL=DABS(H)*EEOET*SCALE/752400.0D0
C
      IF (ESTTOL .LE. 1.0D0) GO TO 260
C
C
C     UNSUCCESSFUL STEP
C                        REDUCE THE STEPSIZE , TRY AGAIN
C                        THE DECREASE IS LIMITED TO A FACTOR OF 1/10
C
      HFAILD= .TRUE.
      OUTPUT= .FALSE.
      S=0.1D0
      IF (ESTTOL .LT. 59049.0D0) S=0.9D0/ESTTOL**0.2D0
      H=S*H
      IF (DABS(H) .GT. HMIN) GO TO 200
C
C     REQUESTED ERROR UNATTAINABLE AT SMALLEST ALLOWABLE STEPSIZE
      IFLAG=6
      KFLAG=6
      RETURN
C
C
C     SUCCESSFUL STEP
C                         STORE SOLUTION AT T+H
C                         AND EVALUATE DERIVATIVES THERE
C
  260 T=T+H
      DO 270 K=1,NEQN
  270   Y(K)=F1(K)
      A=T
      CALL F(A,Y,YP)
      NFE=NFE+1
C
C
C                        CHOOSE NEXT STEPSIZE
C                        THE INCREASE IS LIMITED TO A FACTOR OF 5
C                        IF STEP FAILURE HAS JUST OCCURRED, NEXT
C                           STEPSIZE IS NOT ALLOWED TO INCREASE
C
      S=5.0D0
      IF (ESTTOL .GT. 1.889568D-4) S=0.9D0/ESTTOL**0.2D0
      IF (HFAILD) S=DMIN1(S,1.0D0)
      H=DSIGN(DMAX1(S*DABS(H),HMIN),H)
C
C     END OF CORE INTEGRATOR
C
C
C     SHOULD WE TAKE ANOTHER STEP
C
      IF (OUTPUT) GO TO 300
      IF (IFLAG .GT. 0) GO TO 100
C
C
C     INTEGRATION SUCCESSFULLY COMPLETED
C
C     ONE-STEP MODE
      IFLAG=-2
      RETURN
C
```

```
C     INTERVAL MODE
  300 T=TOUT
      IFLAG=2
      RETURN
C
      END
      SUBROUTINE FEHL(F,NEQN,Y,T,H,YP,F1,F2,F3,F4,F5,S)
C
C     FEHLBERG FOURTH-FIFTH ORDER RUNGE-KUTTA METHOD
C
C    FEHL INTEGRATES A SYSTEM OF NEQN FIRST ORDER
C    ORDINARY DIFFERENTIAL EQUATIONS OF THE FORM
C             DY(I)/DT=F(T,Y(1),---,Y(NEQN))
C    WHERE THE INITIAL VALUES Y(I) AND THE INITIAL DERIVATIVES
C    YP(I) ARE SPECIFIED AT THE STARTING POINT T. FEHL ADVANCES
C    THE SOLUTION OVER THE FIXED STEP H AND RETURNS
C    THE FIFTH ORDER (SIXTH ORDER ACCURATE LOCALLY) SOLUTION
C    APPROXIMATION AT T+H IN ARRAY S(I).
C    F1,---,F5 ARE ARRAYS OF DIMENSION NEQN WHICH ARE NEEDED
C    FOR INTERNAL STORAGE.
C    THE FORMULAS HAVE BEEN GROUPED TO CONTROL LOSS OF SIGNIFICANCE.
C    FEHL SHOULD BE CALLED WITH AN H NOT SMALLER THAN 13 UNITS OF
C    ROUNDOFF IN T SO THAT THE VARIOUS INDEPENDENT ARGUMENTS CAN BE
C    DISTINGUISHED.
C
C
      INTEGER  NEQN
      DOUBLE PRECISION  Y(NEQN),T,H,YP(NEQN),F1(NEQN),F2(NEQN),
     1  F3(NEQN),F4(NEQN),F5(NEQN),S(NEQN)
C
      DOUBLE PRECISION  CH
      INTEGER  K
C...
      EXTERNAL F
      CH=H/4.0D0
      DO 221 K=1,NEQN
  221   F5(K)=Y(K)+CH*YP(K)
      CALL F(T+CH,F5,F1)
C
      CH=3.0D0*H/32.0D0
      DO 222 K=1,NEQN
  222   F5(K)=Y(K)+CH*(YP(K)+3.0D0*F1(K))
      CALL F(T+3.0D0*H/8.0D0,F5,F2)
C
      CH=H/2197.0D0
      DO 223 K=1,NEQN
  223   F5(K)=Y(K)+CH*(1932.0D0*YP(K)+(7296.0D0*F2(K)-7200.0D0*F1(K)))
      CALL F(T+12.0D0*H/13.0D0,F5,F3)
C
      CH=H/4104.0D0
      DO 224 K=1,NEQN
  224   F5(K)=Y(K)+CH*((8341.0D0*YP(K)-845.0D0*F3(K))+
     1                          (29440.0D0*F2(K)-32832.0D0*F1(K)))
      CALL F(T+H,F5,F4)
C
      CH=H/20520.0D0
      DO 225 K=1,NEQN
  225   F1(K)=Y(K)+CH*((-6080.0D0*YP(K)+(9295.0D0*F3(K)-
     1          5643.0D0*F4(K)))+(41040.0D0*F1(K)-28352.0D0*F2(K)))
      CALL F(T+H/2.0D0,F1,F5)
C
C     COMPUTE APPROXIMATE SOLUTION AT T+H
C
      CH=H/7618050.0D0
      DO 230 K=1,NEQN
  230   S(K)=Y(K)+CH*((902880.0D0*YP(K)+(3855735.0D0*F3(K)-
     1          1371249.0D0*F4(K)))+(3953664.0D0*F2(K)+
     2          277020.0D0*F5(K)))
C
```

```
RETURN
END
```

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Anthropometry and Mass Distribution for Human Analogues. USAARL 88-5. v. 1: Military Male Aviators, March 1988.

2. Antonutto, G; Capelli, C; di Prampero, PE. Pedalling in space as a countermeasure to microgravity deconditioning. Microgravity. 1(2): 93-101; 1991.

3. Arduini, C; Baiocco, P; Mortari, D; Parisse, M. A Quasi Adaptive Magnetic Damping Strategy for Gravity Gradient Stabilized Spacecraft. Proceedings of the 43rd IAF, International Astronautical Congress, August 28-September 5, 1992, Washington, D.C. IAF Paper 92-0031: 13 pp; August 1992.

4. Arno, RD; Horkachuck, MJ. Research centrifuge accommodations on space station freedom. Presented at the 20th Intersociety Conference on Environmental Systems, July 9-12, 1990, Williamsburg, VA. Warrendale, PA: Society of Automotive Engineers; 1990: 1-9. ISSN: SAE Paper #901304.

5. Assessment of Programs in Space Biology and Medicine 1991. NASA-CR-190930; 1991: 79 pp.

6. Åstrand, PO; Rodahl, K. Textbook of Work Physiology - Physiological Bases of Exercise, 3rd ed. New York: McGraw-Hill; 1986.

7. Avula, XJR; Ostreicher, HL. Mathematical Model of the Cardiovascular System Under Acceleration Stress. Aviat. Space Environ. Med. 49(1): 279-286; 1978.

8. Barret, C. Spacecraft Flight Control System Design Selection Process for a Geostationary Communication Satellite. NASA Technical Paper NASA-TP-3289; 1992: 20 pp.

9. Benson, AJ. Effect of spaceflight cn thresholds of perception of angular and linear motion. Arch Otorhinolaryngol. 244: 147-154; 1987.

10. Benson, AJ; Brown, SF. Visual display lowers detection threshold of angular, but not linear, whole-body motion stimuli. Aviat. Space Environ. Med. 60(7): 629-633; 1989 July.

11. Benson, AJ; Kass, JR; Vogel, H. European vestibular experiments on the Spacelab-1 mission: 4. Thresholds of perception of whole-body linear oscillation. Exp Brain Res. 64: 264-271; 1986.

12. Benson, AJ; Spencer, MB; Stott, JRR. Thresholds for the detection of the direction of whole-body, linear movement in the horizontal plane . Aviat. Space Environ. Med. 57(11): 1088-1096; 1986 November.

13. Bergel, DH. The Dynamic Elastic Properties of the Arterial Wall. J. Physiol. 156: 458-469; 1961.

14. Berry, P; Berry, I; Manelfe, C. Magnetic Resonance Imaging Evaluation of Lower Limb Muscles During Bed Rest - A Microgravity Simulation Model. Aviat. Space Environ. Med. 64:212-8; 1993.

15. Bioastronautics Data Book. Final Report. Arlington, VA: Office of Naval Research, Physiology Programs. 1972 Sep: 930 pp. ISSN: [PHOTOCOPY EDITION]. WD 750 B616b 1973.

16. Bomar, JB; Pancratz, DJ; Raddin, JH; Harmony, DW; Hessheimer, M; Jacob, JB. Engineering Design Analysis of a Large Radius Track Centrifuge. Final Report for Research Conducted under U.S.A.F. Small Business Innovation Research Contract #F41624-91-C-2002. Submitted for Publication as U.S.A.F. Armstrong Laboratory Technical Report, February, 1993.

17. Bjurstedt, H; Rosenhamer, G; Tyden, G. Acceleration stress and effects of propranolol on cardiovascular responses. Acta Physiol Scand. 90: 491-500; 1974.

18. Bles, W; Bos, JE; Furrer, R; de Graaf, B; Hosman, RJAW; Kortschot, HW; Krol, JR; Kuipers, A; Marcus, JT; Messerschmid, E; Ockels, WJ; Oosterveld, WJ; Smit, J; Wertheim, AH; Wientjes, CJE. Space Adaptation Syndrome Induced by a Long Duration +3Gx Centrifuge Run. IZF 1989-25; AD-A218 248. 1989; 44 pp.

19. Boutellier, U; Arieli, R; Farhi, LE. Ventilation and CO$_2$ response during $+G_z$ acceleration. Respir Physiol. 62: 141-151; 1985.

20. Bungo, MW; Charles, JB; Johnson, PC, Jr. Cardiovascular deconditioning during space flight and the use of saline as a countermeasure to orthostatic intolerance. Aviat. Space Environ. Med. 56: 985-990; 1985.

21. Burton, AC. Physiology and Biophysics of the Circulation. Chicago, IL: Year Book Medical Publishers, Inc.; 1965.

22. Burton, RR. A human-use centrifuge for space stations: proposed ground-based studies. Aviat. Space Environ. Med. 579-582; 1988 June.

23. Burton, RR. Periodic acceleration stimulation in space. Presented at the 19th Intersociety Conference on Environmental Systems, July 24-26, 1989, San Diego, CA. Warrendale, PA: Society of Automotive Engineers; 1989: 1-4. ISSN: SAE Paper #891434.

24. Burton, RR. Physiologic Bases of G-Protection Methods. SAFE Journal. 18(4): 20-24.

25. Burton, RR; Besch, EL; Sluka, SJ; Smith, AH. Differential effects of chronic acceleration on skeletal muscles. J Appl Physiol. 23(1): 80-84; 1967.

26. Burton, RR; MacKenzie, WF. Joint Committee on Aviation Pathology. II. Heart pathology associated with exposure to high sustained +Gz. Aviat. Space Environ. Med. 46(10):1251-1253; 1975 October.

27. Burton, RR; Meeker, LJ; Raddin, JH, Jr. Centrifuges for studying the effects of sustained acceleration on human physiology. IEEE Eng Med Biol. 10: 56-65; 1991 March.

28. Burton, RR; Meeker, LJ. Physiologic validation of a short-arm centrifuge for space application. Aviat. Space Environ. Med. 63: 476-481; 1992 June.

29. Burton, RR; Smith, AH. Adaptation to Acceleration Environments. IN PRESS: Handbook of Physiology - Adaptation to the Environment. Chapter 36. American Physiology Society.

30. Burton, RR; Smith, AH. Hematological findings associated with chronic acceleration. Space Life Sci. 1: 501-513; 1969.

31. Cardus, D; Diamandis, P; McTaggart, WG; Campbell, S. Development of an Artificial Gravity Sleeper (AGS). Physiologist. 33(1 Suppl): S112-S113; 1990 February.

32. Chuong, CJ. Characterization and Modeling of Thoraco-Abdominal Response to Waves. IN: Biomechanical Model of Lung Injury Mechanisms. AD-A190 811. v. 6: 52 pp; 1985 May.

33. Cowing, KL. Possible biomedical applications and limitations of a variable-force centrifuge on the lunar surface: a research tool and an enabling resource. IN: NASA Johnson Space Center, the Second Conference on Lunar Bases and Space Activities of the 21st Century. v. 1: 353-357; 1992 September.

34. Cramer, DB. Physiological considerations of artificial gravity. IN: NASA Marshall Space Flight Center Application of Tethers in Space. v. 13: 13 pp; 1985 March.

35. D'Aunno, DS; Robinson, RR; Smith, GS; Thomason, DB; Booth, FW. Intermittent Acceleration as a Countermeasure to Soleus Muscle Atrophy. J Appl Physiol. 72(2): 428-433; 1992 February.

36. D'Aunno, DS; Thomason, DB; Booth, FW. Centrifugal Intensity and Duration as Countermeasures to Soleus Muscle Atrophy. J Appl Physiol. 69(4): 1387-1389; 1990 October.

37. DeBra, DB; Rodden, J. A Review of Space Guidance and Control Equipment. IN: Automatic Control in Aerospace, IFAC Symposium, July 17-21, 1989, Tsukuba, Japan. Oxford, England: Pergamon Press; 1990; 141-145.

38. Diamandis, PH. Use of a 2-Meter Radius Centrifuge on Space Station for Human Physiologic Conditioning and Testing. Proceedings of the 8th Princeton/AIAA/SSI Conference, "Space Manufacturing 6 - Nonterrestrial Resources, Biosciences, and Space Engineering", May 6-9, 1987, Princeton, NJ. Washington, D.C.: American Institute of Aeronautics and Astronautics; 1987; 133-136.

39. DiZio, P; Lackner, JR; Evanoff, JN. The Influence of Gravitoinertial Force Level on Oculomotor and Perceptual Responses to Coriolis, Cross-Coupling Stimulation. Aviat. Space Environ. Med. 58(9 Suppl):A218-23; 1987.

40. Dodd, KT; Yelverton, JT; Richmond, DR; Morris, JR; Ripple, GR. Nonauditory Injury Threshold for Repeated Intense Freefield Impulse Noise. AD-A224 108. 1990 Mar: 8 pp.

41. Elgersma, M; Stein, G; Jackson, M; Yeichner, J. Robust Controllers for Space Station Momentum Management. IN: 30th IEEE Conference on Decision and Control, December 11-13, 1991, Brighton, United Kingdom. New York, NY: Institute of Electrical and Electronics Engineers; 1991; 2206-2212.

42. Emslie, AG. Active Gradient Torque Control for Earth-Orbiting Astronomical Telescopes. IN: 10th American Control Conference, June 26-26, 1991, Boston, MA. Piscataway, NJ: Institute of Electrical and Electronics Engineers; 1991; 833-838.

43. Fung, YC. Biomechanics. New York: Springer-Verlag; 1993.

44. Gillingham, KK; Burton, RR. Transfer Functions for Arterial Oxygen Saturation During +Gz Stress. Aviat. Space Environ. Med. 46: 1329-1335; 1975 November.

45. Gillingham, KK; Wolfe, JW. Spatial Orientation in Flight. USAFSAM-TR-85-31. 134 pp; 1986 December.

46. Golding, JF; Benson, AJ. Perceptual Saling of Whole-Body Low Frequency Linear Oscillatory Motion. Aviat. Space Environ. Med. 64:636-40; 1993.

47. Greenleaf, JE; Bulbulian, R; Bernauer, EM; Haskell, WL; Moore, T. Exercise-training protocols for astronauts in microgravity. J Appl Physiol. 67(6): 2191-2204; 1989.

48. Grimes, RH; Nyberg, JW; White, WJ. Consequence of Heart-to-Foot Acceleration on Gradient for Tolerance to Positive Acceleration. Aerosp Med. 37: 665-6668; 1966 July.

49. Guedry, FE, Jr; Benson, AJ; Moore, HJ. Influence of a visual display and frequency of whole-body angular oscillation on incidence of motion sickness. Aviat. Space Environ. Med. 53(6): 564-569; 1982 June.

50. Guedry, FE; Oman, CM. Vestibular Stimulation During a Simple Centrifuge Run. Naval Aerospace Medical Research Laboratory, Naval Air Station, Pensacola, FL. NAMRL-1353; (90-10-03-076)AD-A227 285; May 1990.

51. Guyton, AC. Textbook of Medical Physiology. 7th Ed. Philadelphia, PA: W.B. Saunders; 1985.

52. Halstead, TW; Brown, AH; Fuller, CA; Oyama, J. Artificial gravity studies and design considerations for space station centrifuges. Proceedings of the 14th Intersociety Conference on Environmental Systems, July 16-19, 1984, San Diego, CA. Warrendale, PA: Society of Automotive Engineers; 1984; 1-11. ISSN: SAE Paper #840949.

53. Harding, R. Survival in Space. Routledge, London: 1989.

54. Harduvel, JT. Continuous Momentum Management of Earth-Oriented Spacecraft. IN: AIAA Guidance, Navigation and Control Conference, August 20-22, 1990, Portland, OR. Washington, D.C.: American Institute of Aeronautics and Astronautics; 1990; 1-11. ISSN: AIAA Paper 90-3315.

55. Hicks, JW; Badeer, HS. Gravity and the circulation: "open" vs. "closed" systems. Am J Physiol. 262: R725-R732; 1992.

56. Hoffman, EJ. Spacecraft Design Innovations in the APL Space Department. Johns Hopkins APL Technical Digest. 13(1): 167-181; 1992 January.

57. Hubbard, GS; Hargens, AR. Sustaining Humans In Space. Mechanical Engineering. 40-44; 1989 September.

58. Hunt, JW; Ray, JC. Flexible Booms, Momentum Wheels, and Subtle Gravity-Gradient Instabilities. AIAA, Space Programs and Technologies Conference, March 24-27, 1992, Huntsville, Alabama. 1992 Mar; 10 pp. AIAA Paper 92-1673.

59. Jaron, D; Moore, TW; Bai, J. Cardiovascular Response to Acceleration Stress: A Computer Simulation. Proceedings of the IEEE. 76(6): 700-707; 1988.

60. Jaron, D; Moore, TW; Chu, CL. A Cardiovascular Model for Studying Impairment of Cerebral Function During $+G_z$Stress. Aviat. Space Environ. Med. 55(1): 24-31; 1984.

61. Johnson, CC. The Biological Flight Research Facility. Proceedings of the 42nd International Astronautical Congress, October 5-11, 1991, Montreal, Quebec, Canada. 1991 Oct: 8 pp. ISSN: IAF Paper 91-578.

62. Johnson, CC; Hargens, AR. Scientific uses and technical implementation of a variable gravity centrifuge on space station freedom. Presented at the 20th Intersociety Conference on Environmental Systems, July 9-12, 1990, Williamsburg, VA. Warrendale, PA: Society of Automotive Engineers; 1990; 1-9. ISSN: SAE Paper #901360.

63. Jones, GM. Origin Significance and Amelioration of Coriolis Illusions from the Semicircular Canals: A Non-mathematical Appraisal. Aerospace Med. 41(5): 483-490, 1970.

64. Kautz, SA; Feltner, ME; Coyle, EF; Baylor, AM. Pedaling Techniques of Elite Endurance Cyclists: Changes with increasing workload at constant cadence. IN: International Journal of Sport Biomechanics. 7(1): 29-53; 1991 February.

65. Krutz, RW, Jr; Bagian, J; Burton, RR; Meeker, LJ. Heart rate and pulmonary function while wearing the launch entry crew escape suit (LES) during +Gx acceleration and simulated shuttle launch. Proceedings of the 20th Intersociety Conference on Environmental Systems, July 9-12, 1990, Williamsburg, VA. W'arrendale, PA: Society of Automotive Engineers; 1990; 1-3. ISSN: SAE Paper #901358.

66. Longman, RW. The Kinetics and Workspace of a Robot Mounted on a Satellite That Is Free to Rotate and Translate. Aeronautics and Astronautics. 374-381; 1988.

67. Ormsby, CC; Young, LR. Perception of Static Orientation in a Constant Gravitoinertial Environment. Aviat. Space Environ. Med. 47(2): 159-164; 1976.

68. Man-Systems Integration Standards. NASA-STD 3000. Rev. A; v. 1; 1989 October.

69. Meeker, LJ; Isdahl, W. Parametric Design Study for a Small Radius Centrifuge for Space Application. Presented at the 10th IAA Man in Space Symposium, April 1993, Tokyo, Japan.

70. Mergner, T; Rottler, G; Kimmig, H; Becker, W. Role of Vestibular and Neck Inputs for the Perception of Object Motion in Space. Exp Brain Res. 89(3): 655-668; 1992.

71. Meriam, JL. Engineering Mechanics, Statics and Dynamics. New York: John Wiley & Sons, Inc.; 1978.

72. Murakami, DM; Fuller, CA. The Effect of Hyperdynamic Fields on the Oxidative Metabolism of thr Paraventricular Nucleus. Aviat. Space Environ. Med. 61(8): 722-724; 1990 August.

73. Newsom, BD; Brady, JF; Stumm, JE. Gyroscopic Stimulation of the Semicircular Canals During Sensory Deprivation. Aerospace Med. 42(12): 1283-1289; 1971.

74. Oman, CM. A heuristic mathematical model for the dynamics of sensory conflict and motion sickness. Acta Otolaryngol Suppl. 392: 1-44; 1982.

75. Oman, CM; Lichtenberg, BK; Money, KE. Space motion sickness monitoring experiment: Spacelab 1. IN: AGARD Motion Sickness: Mechanisms, Prediction, Prevention and Treatment. 1984 November; 21 pp.

76. Piemme, TE; Hyde, AS; McCally, M; Potor, G. Jr. Human tolerance to G_z 100 per cent gradient spin. Aerospace Med. 37: 16-21; 1966 January.

77. Psiaki, ML; Martel, F; Pal, PK. Three-Axis Attitude Determination via Kalman Filtering of Magnetometer Data. J Guidance Control Dynamics. 13(1): 506-514; 1990 May.

78. Quadrelli, BM; Lorenzini, EC. Dynamics and Stability of a Tethered Centrifuge in Low Earth Orbit. J Astronaut Sci. 40: 3-5; 1992 January.

79. Raddin, JH, Jr; Scott, WR; Ziegler, J; Benedict, JV; Smith, HL. Concept Feasibility Analysis for a Large Radius Track-Centrifuge. Final Report for Research Conducted under U.S.A.F. Small Business Innovation Research Contract #F41622-89-C-1025. Submitted for Publication as U.S.A.F. School of Aerospace Medicine Technical Report, February, 1990.

80. Raphan, T; Dai, M; Cohen, B. Spatial Orientation of the Vestibular System. Ann N Y Acad Sci. 656: 140-157; 1992 May 22.

81. Raphan, T; Sturm, D. Modeling the Spatiotemporal Organization of Velocity Storage in the Vestibuloocular Reflex by Optokinetic Studies. J Neurophysiol. 66(4): 1410-1421; 1991 October.

82. Ratino, DA; Repperger, DW; Goodyear, C; Potor, G; Rodriquez, LE. Quantification of Reaction Time and Time Perception During Space Shuttle Operations. AAMR-TR-88-018. 1988 Apr 12; 5 pp.

83. Reason, JT; Benson, AJ. Voluntary movement control and adaptation to cross-coupled simulation. Aviat. Space Environ. Med. 49(11): 1275-1280; 1978 November.

84. Rideout, VC; Dick, DE. Difference-Differential Equations for Fluid Flow in Distensible Tubes. IEEE Trans Bio-Med Eng. 14(4): 171-177; 1967.

85. Schiesser, WE. The Numerical Method of Lines. San Diego: Academic Press Inc.; 1991.

86. Searby, N. Effect of science laboratory centrifuge of space station environment. IN: NASA, Marshall Space Flight Center, Measurement and Characterization of the Acceleration Environment on Board the Space Station. 1990 Aug; 17 pp.

87. Shulzhenko, EB; Vil-Viliams, IF. Short radius centrifuge as a method of long-term space flights. Physiologist. 35(1 Suppl): 5122-5125; 1992.

88. Snyder, MF; Rideout, VC. Computer Simulation Studies of the Venous Circulation. IEEE Trans. Bio-Med Eng. 16(4): 325-334; 1969 October.

89. Stone, HL; Lindsey, JN; Sordahl, LA; Erickson, HH; Dowell, RT. The Pathophysiology of High Sustained +Gz Acceleration, Limitation to Air Combat Maneuvering and the Use of Centrifuges in Performance Training. AFOSR-TR-77-1038. 1976; 8 pp.

90. Stuhmiller, JH. Modeling of the Non-Auditory Response to Blast Overpressure. Considerations in Developing a Mechanistically-Based Model of Blast-Induced Injury to Air-Containing Organs. AD-A223 585. 1990 January; 26 pp.

91. Vander Vorst, MJ; Stuhmiller, JH. Modeling of the Non-Auditory Response to Blast Overpressure. Calculation of the Internal Mechanical Response of Sheep tp Blast Loading. AD-A223 392. 1990 January; 64 pp.

92. Vander Vorst, MJ; Stuhmiller, JH. Modeling of the Non-Auditory Response to Blast Overpressure. Calculation of Parenchymal Pressure Due to Double Peak Loading. AD-A223 664. 1990 January: 20 pp.

93. Vil'-Vil'yams, IF. Comparative Characteristics of Human Resistance to Long-Term Overloads with Gravitational Gradients. Hum Physiol. 7(1): 41-44; 1981 January.

94. Voge, VM. Simulator Sickness Provoked by A Human Centrifuge. Milit Med. 156(10): 575-577; 1991 October.

95. White, RJ; Cronton, DG, Fitzgerald, DG. Cardiovascular Modelling: Simulating the Human Cardiovascular Response to Exercise, Lower Body Negative Pressure, Zero Gravity and Clinical Conditions. Adv. Cardiovasc. Phys. Part 1: 195-299; 1983.

96. White, WJ; Nyberg, JW; White, PD; Grimes, RH; Finney, LM. Biomedical Potential of a Centrifuge in an Orbiting Laboratory. SM-48502; SSD-TDR-64-209-Suppl: 129 pp. 1965 July.

97. Wilcox, DC. Numerical Algorithm for Rapid Integration of Turbulence Model Equations on Elliptic Regions. AFOSR-TR-82-0213. 1982 February; 26 pp.

98. Williams, CE. Attitude Determination of Triad and Tip-II and -III Gravity-Gradient-Stabilized Satellites. APL/JHU/TG-1313. 1977 December; 63 pp.

99. Womersley, JR. An elastic tube theory of pulse transmission and oscillatory flow in mammalian arteries. Wright Air Development Report WADC-TR-56-614. 1957.

100. Young, LR. Modelling Human Disorientation in a Rotating Spacecraft. American Institute of Aeronautics and Astronautics; AIAA/ASMA Weightlessness and Artificial Gravity Meeting, August 9-11, 1971, Williamsburg, Virginia. A71-366 54.

101. Young, LR; Oman, CM; Watt, DGD; Money, KE; Lichtenberg, BK; Kenyon, RV; Arrott, AP. M.I.T./Canadian vestibular experiments in Spacelab-1 mission: 1. Sensory adaptation to weightlessness and readaptation to one-g: an overview. Exp Brain Res. 64: 291-298; 1986.

102. Yu, JH; Ho, HH; Stuhmiller, JH. Modeling of the Non-Auditory Response to Blast Overpressure. A Surrogate Model of Thoracic Response to Blast Loading. AD-A223 393. 1990 January; 43 pp.

103. Zacharias, GL. Motion Cue Models for Pilot-Vehicle Analysis. Technical Report for Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio; Mar 1978. AMRL Technical Report 78-2, Contract F33615-77-C-0506, January 1-September 30, 1977. Cambridge, MA: Bolt Eranek and Newman, Inc., Control Systems Department; 1978 March.

104. Zborovskaya, BI. Use of short-arm centrifuge to prevent deconditioning when immersed in water. USSR Rep Space Biol Aerospace Med. 13(5): 114-116; 1979 October.

☆U.S. GOVERNMENT PRINTING OFFICE: 1994 - 560-051/00001